

(18)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-75858

(43)公開日 平成6年(1994)3月18日

(51)Int. Cl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 12/08	3 1 0 Z	7608-5B		
11/28	3 1 0 B	9280-5B		
// G 0 6 F 9/32	3 2 0 F	9189-5B		

審査請求 有 請求項の数13(全 28 頁)

(21)出願番号 特願平5-128091

(22)出願日 平成5年(1993)4月30日

(31)優先権主張番号 特願平4-119403

(32)優先日 平4(1992)5月13日

(33)優先権主張国 日本 (J P)

(31)優先権主張番号 特願平4-187497

(32)優先日 平4(1992)6月22日

(33)優先権主張国 日本 (J P)

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72)発明者 新井 智久

東京都港区芝五丁目7番1号 日本電気株式会社内

(72)発明者 堀口 由美子

東京都港区芝五丁目7番1号 日本電気株式会社内

(74)代理人 弁理士 五十嵐 省三

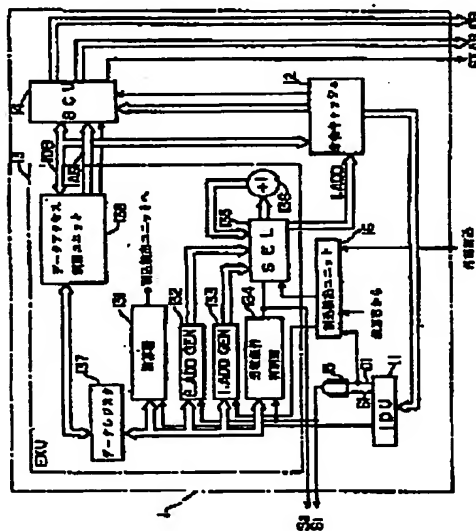
(54)【発明の名称】 キャッシュ内蔵マイクロプロセッサ及びそのトレースシステム

(57)【要約】

【目的】 キャッシュ内蔵マイクロプロセッサにおいて効率のよいトレース方法を行えるようにすること。

【構成】 分岐命令が実行された否かを示す信号S<sub>1</sub>及び分岐命令の実行後の分岐成立／不成立を示す信号S<sub>2</sub>をマイクロプロセッサ1の外部へ送出する。これらの信号のもとづいての命令のトレースを行う。

本発明の第1の実施例



## 【特許請求の範囲】

【請求項1】 実行すべき分岐命令を検出する分岐命令検出手段と、

前記分岐命令が検出されたか否かを外部に通知する分岐命令検出通知手段と、

前記分岐命令の実行によって分岐が成立したか否を判別する分岐成立／不成立判別手段と、

前記分岐が成立したか否かを外部に通知する分岐成立／不成立通知手段とを具備するキャッシュ内蔵マイクロプロセッサ。

【請求項2】 前記各通知手段は外部ステータス信号端子を用いる請求項1に記載のキャッシュ内蔵マイクロプロセッサ。

【請求項3】 さらに、

前記分岐命令が分岐先を予め静的に決定できない動的な分岐命令であるか否かを判別する動的な分岐命令判別手段と、

前記動的な分岐命令が検出されたときに割込みを発生する割込発生手段と、

該割込みが発生したときにトレーストラップ処理を実行して前記分岐先を演算して外部に出力するトレーストラップ処理手段とを具備する請求項1に記載のキャッシュ内蔵マイクロプロセッサ。

【請求項4】 さらに、命令キャッシュ(12)が活性化されているか否かを判別する命令キャッシュ活性判別手段と、

前記命令キャッシュが活性化しているか否かを外部に通知するキャッシュ活性／非活性通知手段とを具備する請求項1に記載のキャッシュ内蔵マイクロプロセッサ。

【請求項5】 前記各通知手段は外部ステータス信号端子(ST)を用いる請求項4に記載のキャッシュ内蔵マイクロプロセッサ。

【請求項6】 命令アドレスを発生する実行ユニット(13)と、

該実行ユニットに接続され、前記命令アドレスを受信し、該命令アドレスが登録されているときには該命令アドレスに対応する命令コードを発生し、前記命令アドレスが登録されていないときにはリフレッシュ要求を外部へ発生する命令キャッシュ(12)と、

該命令キャッシュ及び前記実行ユニットに接続され、前記命令キャッシュからの命令コードをデコードして分岐命令を含む制御信号を前記実行ユニットに発生する命令デコーダ(11)と、

前記命令デコーダに接続され、前記分岐命令の発生を外部に出力する分岐命令発生出力手段と、

前記実行ユニットに接続され、該実行ユニットが前記分岐命令を実行して分岐が成立したときに該分岐の成立を外部へ出力する分岐命令成立出力手段とを具備するマイクロプロセッサ。

【請求項7】 さらに、

前記実行ユニット及び前記命令キャッシュに接続され、前記マイクロプロセッサの状態を示すステータス信号を外部へ出力するバス制御ユニット(14)を具備し、前記分岐命令発生出力手段及び前記分岐命令成立出力手段は前記マイクロプロセッサのステータス信号端子を用いる請求項6に記載のマイクロプロセッサ。

【請求項8】 さらに、

前記分岐命令が分岐先を予め静的に決定できない動的な分岐命令のときに割込みを発生する割込発生手段を具備し、

前記実行ユニットは前記割込みを受信してトレーストラップ処理を実行し、前記分岐先を外部へ出力する出力手段を具備する請求項6に記載のマイクロプロセッサ。

【請求項9】 さらに、

前記実行ユニット及び前記命令キャッシュに接続され、該命令キャッシュを活性化／非活性化し該命令キャッシュの活性／非活性に応じてキャッシュオン／オフ信号を外部へ出力するキャッシュ制御レジスタ(18)を具備する請求項7に記載のマイクロプロセッサ。

【請求項10】 実行すべき分岐命令を検出したか否かを示す検出信号(S1)、該分岐命令を実行したときに分岐が成立したか否かを示す分岐成立／非成立信号(S2)及び外部メモリ(2)にアクセスを要求する命令アドレスを伴うアクセス信号を出力する、命令キャッシュ(13)内蔵マイクロプロセッサ(1)のトレースシステムであって、

前記検出信号を受信して分岐命令が実行されたか否かを判別する手段と、

前記分岐成立／非成立信号を受信して分岐が成立したか否かを判別する手段と、

前記アクセス信号を受信して前記外部メモリにアクセスされたか否かを判別する手段と、

分岐命令が実行されかつ前記外部メモリにアクセスされたときに、前記出力された命令アドレスをトレースする手段と、

分岐命令が実行され、前記分岐命令の実行後分岐が不成立し、かつ前記外部メモリにアクセスされていないときに、次のブロックをトレースする手段と、

分岐命令が実行され、前記分岐命令の実行後分岐が成立し、かつ前記外部メモリにアクセスされていないときに、分岐先のブロックをトレースする手段とを具備するトレースシステム。

【請求項11】 実行すべき分岐命令を検出したか否かを示す検出信号(S1)、該分岐命令を実行したときに分岐が成立したか否かを示す分岐成立／不成立信号(S2)及び外部メモリ(2)にアクセスを要求する命令アドレスを伴うアクセス信号を出力し、前記分岐命令が予め静的に決定できない動的な分岐命令であるときにトレーストラップ処理を実行して分岐先アドレスを外部へ出力する、命令キャッシュ(13)内蔵マイクロプロセッサ

サ(1)のトレースシステムであって、  
 前記検出信号を受信して分岐命令が実行されたか否かを判別する手段と、  
 前記分岐成立／不成立信号を受信して分岐が成立したか否かを判別する手段と、  
 前記アクセス信号を受信して前記外部メモリにアクセスされたか否かを判別する手段と、  
 分岐命令が実行されかつ前記外部メモリにアクセスされたときに、前記出力された命令アドレスをトレースする手段と、  
 分岐命令が実行され、前記分岐命令の実行後分岐が不成立し、かつ前記外部メモリにアクセスされていないときに、次のブロックをトレースする手段と、  
 分岐命令が実行され、前記分岐命令の実行後分岐が成立し、前記外部メモリにアクセスされず、かつトレーストラップ処理が実行されていない時に、分岐先のブロックをトレースする手段と、  
 分岐命令が実行され、前記分岐命令の実行後分岐が成立し、前記外部メモリにアクセスされず、かつトレーストラップ処理が実行されているときに、該トレーストラップ処理によって出力された分岐先アドレスをトレースする手段とを具備するトレースシステム。  
 【請求項12】 実行すべき分岐命令を検出したか否かを示す検出信号(S<sub>1</sub>)、該分岐命令を実行したときに分岐が成立したか否かを示す分岐成立／不成立信号(S<sub>2</sub>)、外部メモリ(2)にアクセスを要求する命令アドレスを伴うアクセス信号及び命令キャッシュ(13)を活性化／非活性化するキャッシュオン／オフ信号(S<sub>3</sub>)を出力する、命令キャッシュ内蔵マイクロプロセッサ(1)のトレースシステムであって、  
 前記検出信号を受信して分岐命令が実行されたか否かを判別する手段と、  
 前記分岐成立／不成立信号を受信して分岐が成立したか否かを判別する手段と、  
 前記アクセス信号を受信して前記外部メモリにアクセスされたか否かを判別する手段と、  
 前記キャッシュオン／オフ信号を受信して前記命令キャッシュが活性化しているか非活性化しているかを判別する手段と、  
 前記命令キャッシュが非活性化されているとき、もしくは分岐命令が実行されかつ前記外部メモリにアクセスされたときに、前記出力された命令アドレスをトレースする手段と、  
 前記命令キャッシュが活性化され、分岐命令が実行され、前記分岐命令の実行後分岐が不成立し、かつ前記外部メモリにアクセスされていないときに、次のブロックをトレースする手段と、  
 前記命令キャッシュが活性化され、分岐命令が実行され、前記分岐命令の実行後分岐が成立し、前記外部メモリにアクセスされず、かつトレーストラップ処理が実行されているときに、該トレーストラップ処理によって出力された分岐先アドレスをトレースする手段とを具備するトレースシステム。

クをトレースする手段とを具備するトレースシステム。  
 【請求項13】 実行すべき分岐命令を検出したか否かを示す検出信号(S<sub>1</sub>)、該分岐命令を実行したときに分岐が成立したか否かを示す分岐成立／不成立信号(S<sub>2</sub>)、外部メモリ(2)にアクセスを要求する命令アドレスを伴うアクセス信号及び命令キャッシュ(13)を活性化／非活性化するキャッシュオン／オフ信号(S<sub>3</sub>)を出力し、前記分岐命令が予め論理的に決定できない動的な分岐命令であるときにトレーストラップ処理を実行して分岐先アドレスを外部へ出力する、命令キャッシュ内蔵マイクロプロセッサ(1)のトレースシステムであって、  
 前記検出信号を受信して分岐命令が実行されたか否かを判別する手段と、  
 前記分岐成立／不成立信号を受信して分岐が成立したか否かを判別する手段と、  
 前記アクセス信号を受信して前記外部メモリにアクセスされたか否かを判別する手段と、  
 前記キャッシュオン／オフ信号を受信して前記命令キャッシュが活性化されているか非活性化されているかを判別する手段と、  
 前記命令キャッシュが非活性化されているとき、もしくは分岐命令が実行され、かつ前記外部メモリにアクセスされたときに、前記出力された命令アドレスをトレースする手段と、  
 前記命令キャッシュが活性化され、分岐命令が実行され、前記分岐命令の実行後分岐が不成立し、かつ前記外部メモリにアクセスされていないときに、次のブロックをトレースする手段と、  
 前記命令キャッシュが活性化され、分岐命令が実行され、前記分岐命令の実行後分岐が成立し、前記外部メモリにアクセスされず、かつトレーストラップ処理が実行されていないときに、分岐先のブロックをトレースする手段と、  
 前記命令キャッシュが活性化され、分岐命令が実行され、前記分岐命令の実行後分岐が成立し、前記外部メモリにアクセスされず、かつトレーストラップ処理が実行されているときに、該トレーストラップ処理によって出力された分岐先アドレスをトレースする手段とを具備するトレースシステム。  
 【発明の詳細な説明】  
 【0001】  
 【産業上の利用分野】 本発明はシステムもしくはプログラム開発におけるデバッグの容易化を図ったキャッシュを内蔵するマイクロプロセッサ及びそのトレースシステムに関する。  
 【0002】  
 【従来の技術】 マイクロプロセッサのシステムもしくはプログラムの開発においては、デバッグ方法として、トレース、トラップ、シングルステップの3つのレベルの方法がある。

## 【0003】(1) トレース方法

プログラムの実行位置を示す情報を収集してこれらを組立てプログラムの実行順序を認識する。あるいはメモリオペランドのアクセスに関する情報を収集してこれらを組立てメモリオペランドのアクセス順序を認識する。このような情報は、一般に、バスサイクルにおいてマイクロプロセッサの外部端子で観測可能なアドレス信号、データ信号及びステータス信号によって構成できる。このように、上述の情報はマイクロプロセッサの外部で観測できるので、マイクロプロセッサに特別な機能を付加することなく、トレース機能を実現できる。また、デバックすべきプログラムの実行を中断する割込みもしくは例外を発生させる必要がないので、あるいはデバックすべきプログラムの実行に対して特別なバスサイクルを待たせる必要がないので、デバック時の動作タイミングは非デバック時の動作タイミングと同一である。しかし、このトレース方法は、特定の状態を検出してプログラムの実行を中断しないので、デバックすべきプログラムの状態を単に観測するにすぎない受動的な方法である。

## 【0004】(2) トラップ方法

あらかじめ特定された位置(アドレス)において命令コードもしくはメモリオペランドのアクセスが発生すると、デバックと呼ばれるデバックプログラムに制御が移行し、これにより、さらに詳細なデバックを実行する。トレース方法と異なり、必要なタイミングでのみ制御がデバックに移行する。従ってトラップ割込み機構と呼ばれるハードウェア機構を必要とし、これによりあらかじめ設定したアドレスを有する命令コードの実行の直前もしくは直後に割込み/例外を正確に発生する。この結果、トラップ方法は常に実現できない。また、プログラムのシーケンスを正確に知ることができない。さらに、制御があらかじめ定められたトラップにかからなければ、たとえば、プログラムがトラップにかかる前に暴走したときは、プログラムの実行は中断できない。

## 【0005】(3) シングルステップ方法

1命令を実行する毎に、制御をデバックに移行し、これより、マイクロプロセッサの内部状態、たとえば汎用レジスタの状態、プロセッサステータスワード(PSW)、プログラムカウンタ(PC)等を表示したり、また、これらの状態の一部を変更する。このように、マイクロプロセッサの内部状態は1命令の実行毎に認識されるので、プログラムの手順は正確に認識できる。

【0006】このように、シングルステップ方法においては、マイクロプロセッサが1命令を実行する毎に制御をデバックに移行する。従って、シングルステップ割込み機構と呼ばれるハードウェア機構を必要とし、これにより、1命令の実行タイミング毎に割込み/例外を発生する。この結果、シングルステップは常に実現できるとは限らない。さらに、各命令の実行毎に制御をデバックに移行するので、

- ① プログラムの実行効率が低い、
- ② 時間割込みルーチンのようなタイミングクリティカルな処理のデバックは不可能である、
- ③ デバック時における内部動作タイミングが非デバック時における動作タイミングと全く異なる。

【0007】上述のごとく、3つのデバック方法は利点及び欠点を有するので、一般に、これらのデバック方法の組合せを利用する。たとえば、始めに、トレース方法によって命令の実行のシーケンスを認識する。第2に、制御が問題が発生している位置(アドレス)付近にきたときに、制御をトラップ方法によってデバックに移行する。最後に、命令の実行状態をシングルステップ方法によって詳細に追う。このように、トレース方法はシステムもしくはプログラムを効率的にデバックするのに必須である。

【0008】次に、従来のキャッシュ内蔵マイクロプロセッサ及びトレースシステムについて図22及び図23を参照して説明する。図22に示すごとく、従来のマイクロプロセッサ1は命令デコード(IDU)11を含み、この命令デコード11は命令キャッシュ12からの命令コードをデコードして種々の制御信号を実行ユニット(EXU)13に送出する。この結果、実行ユニット13はこれら制御信号に従って種々の処理を実行する。同時に、実行ユニット13は次に実行すべき命令のアドレス(1, ADD)を命令キャッシュ12に送出する。あるいは、実行ユニット13は命令アドレスを内部アドレスバスIABを介してバス制御ユニット(BCU)14に送出し、このアドレスを外部アドレスバスABを介して外部メモリ2(図22に図示せず、図23に図示)に送出することもできる。この場合、バス制御ユニット14は命令フェッチサイクルを示すステータス信号をステータス信号線STに送出し、この結果、外部メモリ2が上述のアドレスにてアクセスされることになる。次に、外部メモリ2からの命令コードがバス制御ユニット14によって受信され、内部データバスIDBを介して命令キャッシュ12に書き込まれることになる。

【0009】外部メモリ2を伴う図22のマイクロプロセッサ1のための従来のトレースシステムは、図23に示すごとく、トレースアナライザ31及びトレースメモリ32よりなるトレーサ3を含む。トレースアナライザ31はステータス線STを監視して命令フェッチサイクルを検出する。トレースメモリ32はトレースアナライザ31からトレースアドレス(T, ADD)を受信して命令のトレースを形成する。

【0010】まず、命令キャッシュ12が動作していない場合のトレーサ3の動作を説明する。つまり、マイクロプロセッサ1に命令フェッチサイクルが発生すると、マイクロプロセッサ1は命令フェッチサイクルを起動してステータス信号線STにそのサイクルを示すステータス信号を送出すると共に、外部アドレスバスABにフェ

ツチすべき命令のアドレスを送出する。この状態において、トレースアナライザ31がこの命令フェッチサイクルを検出すると、トレースアナライザ31はストローブ信号SRをトレースメモリ32に送出し、この結果、命令フェッチサイクルで発生した外部アドレスバスAB上のアドレスがトレースメモリ32に書き込まれることになる。このように、マイクロプロセッサ1によって実行された命令のアドレスはすべてトレースメモリ32に記録され、これにより、命令のトレースを形成できる。

【0011】次に、命令キャッシュ12が動作している場合のトレース3の動作を説明する。この場合にも、実行ユニット13は命令のアドレスを命令キャッシュ12に送出する。この命令が命令キャッシュ12に登録されていないときには（以後、ミスヒットとする）、命令キャッシュ12はミスヒットした命令のフェッチをバス制御ユニット14に要求する。次に、バス制御ユニット14はステータス信号線STにリブレースサイクルを起動するステータス信号を送出し、同時に、外部アドレスバスABにこのミスヒットした命令のアドレスを発生する。この結果、ミスヒットした命令のコードが外部メモリ2から読み出され、命令キャッシュ12に登録されることになる。また、このミスヒットした命令のコードは命令キャッシュ12から命令デコーダ11に供給される。この場合においても、トレースアナライザ31はステータス信号線STに命令フェッチサイクルを検出するので、命令のトレースをトレースメモリ32に形成し、トレース方法を実行できる。このように、キャッシュメモリ内蔵のマイクロプロセッサにおいては、実行すべき命令がキャッシュメモリに登録されていないときに（つまり、ミスヒット）、このミスヒットの命令は外部メモリからフェッチされ、この結果、トレース方法を実行するのに何ら問題ない。

#### 【0012】

【発明が解決しようとする課題】しかしながら、上述の命令が命令キャッシュ12に登録されているときには（以後、ヒットとする）、命令キャッシュ12はヒットされた命令のコードを直ちに命令デコーダ12に送出する。従って、この場合、マイクロプロセッサ1は何ら外部信号を発生しないので、トレース3はトレース方法を実行できないことになる。

【0013】なお、トレース方法を完全に実行するために、トレース時にはキャッシュメモリを非活性にしてフェッチされたすべて命令をマイクロプロセッサの外部から観測可能にする方法が考えられる。しかしながら、この場合には、トレース時のマイクロプロセッサの動作が実際の動作時のマイクロプロセッサの動作と異なってしまうので、正確なデバッグは不可能であるという課題がある。

【0014】従って、本発明の目的は、トレース方法を効率的に行うことを可能にしたキャッシュメモリ内蔵マ

イクロプロセッサを提供することにある。他の目的は、上記キャッシュメモリ内蔵マイクロプロセッサ用のトレースシステムを提供することにある。

#### 【0015】

【課題を解決するための手段】上述の課題を解決するために本発明によれば、キャッシュ内蔵マイクロプロセッサに、実行すべき分岐命令を検出する分岐命令検出手段と、分岐命令が検出されたか否かを外部に通知する分岐命令検出通知手段と、分岐命令の実行によって分岐が成立したか否かを判別する分岐成立／不成立判別手段と、分岐が成立したか否かを外部に通知する分岐成立／不成立通知手段とを設ける。

【0016】また、トレースシステムは、マイクロプロセッサからの分岐命令検出信号及び分岐成立／不成立信号を受け、分岐命令が実行されかつ分岐が不成立したときには次のブロックのトレースを行う手段と、分岐命令が実行されかつ分岐が成立したときには分岐先のブロックを行う手段とを備えている。

#### 【0017】

【作用】上述のマイクロプロセッサにおいて、たとえキャッシュがヒットしても、分岐命令が実行されると、分岐命令検出及びその分岐命令の成立／不成立が外部へ通知される。この結果、トレースシステムはキャッシュのヒット時のマイクロプロセッサの命令実行をトレースできる。

#### 【0018】

【実施例】図1は本発明のキャッシュ内蔵マイクロプロセッサの第1の実施例を示すブロック回路図である。図1において、15はオア回路であって、静的分岐命令S1及び動的分岐命令D1を検出するものである。なお、静的分岐命令が実行されて分岐が成立すれば、その分岐先に制御が移行するが、この場合、この分岐先は予め決定されている。他方、動的分岐命令が実行されて分岐が成立すれば、やはり、その分岐先に制御が移行するが、この場合の分岐先は予め動的に計算つまり決定されていない。動的分岐命令の一例はレジスタ間接分岐命令である。このようにオア回路15が実行される分岐命令を検出する検出信号S1を発生してマイクロプロセッサ1の外部へ直接出力する。

【0019】また、16は動的分岐命令、実行ユニット13からの内部割り込み、及び外部割り込みを検出する割り込み検出ユニットである。実行ユニット13は、内部割り込みを発生する演算器131、分岐先（アドレス）を発生する分岐アドレス発生器132、割り込みアドレスを発生する割り込みアドレス発生器133、及び分岐命令の実行の結果、分岐が成立したか否かを判別する分岐条件判別器134を含む。つまり、分岐が成立すると、セレクト135は分岐アドレスを選択して命令アドレスとして命令キャッシュ12に送出する。また、割り込みが発生すると、セレクト135は割り込みアドレ

スを選択して命令アドレスとして命令キャッシュ12に送出する。逆に、分岐が不成立かつ割り込みも生じないときには、セクタ135は+1インクレメンタ136の出力を選択して命令キャッシュ12に送出する。+1インクレメンタ136はセクタ135の出力を+1増大させるものであり、これにより、順次アクセスを可能とする。

【0020】また、分岐条件判別器134は、分岐命令の実行の結果、分岐が成立したか否かを示す分岐成立／不成立信号S<sub>2</sub>を発生してマイクロプロセッサ1の外部へ直接送出する。さらに、実行ユニット13は、演算器131、分岐アドレス発生器132、割り込みアドレス発生器133及び分岐条件判別器134に接続されたデータレジスタ137、及びこのデータレジスタ137とバス制御ユニット14との間に接続されたデータアクセス制御ユニット138を含む。

【0021】さらに、割り込み検出ユニット16が動的な分岐命令を検出したときには、トレーストラップ処理が起動される。たとえば、始めに、プログラムカウンタ(PC)及びプログラムステータスワード(PSW)（図示せず）が退避される。次に、割り込み検出ユニット16が割り込みを割り込みアドレス発生器133に送出し、また、セクタ135が割り込みアドレス発生器133を選択し、これにより、トレーストラッププログラムをフェッチする。この場合、このフェッチアクセスに対するトレースは実行されない。次に、命令キャッシュ12はフラッシュつまり無効化される。最後に、動的な分岐命令を再び実行する。この場合、分岐条件判別器134が分岐は成立したと判別したときに、分岐アドレス発生器132の分岐先(アドレス)がセクタ135によって選択され、この結果、制御はこの分岐先にジャンプする。この場合、命令キャッシュ12が無効化されており、従って、命令キャッシュ12は必ずミスヒットするので、この分岐先アドレスのトレースが実行されることになる。

【0022】図2は図1の命令キャッシュ12の詳細なブロック回路図である。図2において、命令キャッシュ12は実行ユニット13から供給される命令アドレス(1. ADI)の下位ビットによって同時にアクセスされるタグ部121及びデータ部122を含む。この場合、タグ部121は命令アドレスの上位ビットHBを記憶し、他方、データ部122は命令コードを記憶するが、これらの命令コードは実行ユニット13からの命令アドレスの下位ビットLB及びタグ部121に記憶されている上位ビットHBの組合せの命令アドレスによって指定される。この場合、実行ユニット13からの命令アドレスの上位ビットHBはタグ部121の上位ビットHBとは必ずしも一致しない。従って、実行ユニット13からの命令アドレスの上位ビットHBが実行ユニット13からの命令アドレスの下位ビットLBによってアクセ

スされたタグ部121の上位ビットHBと一致したときに、命令キャッシュ12が“ヒット”したと言える。逆に、実行ユニット13からの命令アドレスの上位ビットHBが実行ユニット13からの命令アドレスの下位ビットLBによってアクセスされたタグ部121の上位ビットHBと一致しないときに、命令キャッシュ12が“ミスヒット”したと言える。

【0023】命令キャッシュ12がヒットしたかミスヒットしたかを判別するために、比較器123を設け、実行ユニット13からの上位ビットHBとタグ部121からの上位ビットHBとを比較している。比較器123のヒット／ミスヒット信号はキャッシュ制御ユニット124に供給される。命令キャッシュ12がヒットすると、キャッシュ制御ユニット124は制御信号をセクタ125に送出し、これにより、セクタ125はデータ部122の出力を選択する。この結果、命令コードがデータ部122から命令デコーダ11に供給されることになる。逆に、命令キャッシュ12がミスヒットすると、キャッシュ制御ユニット124は命令アクセス要求をバス制御ユニット14に送出すると共に、ラッチ／ゲート回路126を制御して実行ユニット13からの命令アドレスをバス制御ユニット14に送出する。従って、バス制御ユニット14はリプレースバスサイクルを起動して外部メモリ2（図5参照）から命令コードをフェッチし、この結果、このフェッチされた命令コードがデータ部122に書き込まれる。同時に、キャッシュ制御ユニット124はセクタ125を制御して内部データバスIDBを選択し、上述のフェッチされた命令コードは命令デコーダ11に供給されることになる。

【0024】図3は図1のバス制御ユニット14の詳細なブロック回路図である。図3において、バス制御ユニット14は、内部データバスIDBと外部データバスDBとの間に設けられたデータバッファ141、及び内部アドレスバスIABと外部アドレスバスABとの間に設けられたアドレスバッファ142を含む。また、アドレスバッファ142は命令キャッシュ12からの命令アドレスを受信する。これら2つのバッファ141、142はバスアクセス制御ユニット143によって制御され、このバスアクセス制御ユニット143は実行ユニット13のデータアクセス制御ユニット138からの制御信号及び命令キャッシュ12からの制御信号を受信する。また、バスアクセス制御ユニット143はステータス(ST)信号、ライト／リード(W/R)信号等が発生する。図1において用いられる命令コードの一例は、図4に示すように、機能フィールド及びレジスタ指定フィールドよりなり、さらに、機能フィールドは、静的な分岐命令ビット、動的な分岐命令ビット等によって形成されている。

【0025】次に、図1のマイクロプロセッサのトレースシステムについて図5、図6を参照して説明する。図

5に示すように、トレースアナライザ31がマイクロプロセッサ1から検出信号S<sub>i</sub>及び分岐成立/不成立信号S<sub>j</sub>を受信する点で図23と異なる。この結果、トレースアナライザ31は図6に示すごとく構成される。

【0026】図6において、トレース制御ユニット311はマイクロプロセッサ1からの検出信号S<sub>i</sub>及びステータスデコーダ313からの制御信号を受信してトレースの起動を示すストローブ信号SRを発生する。たとえば、ステータス信号線STのステータス信号が命令アクセスのバスサイクルを示していれば、ステータスデコーダ313はこのような制御信号を発生する。

【0027】また、図6においては、フロー解析ユニット312はマイクロプロセッサ1から分岐成立/不成立信号S<sub>j</sub>及びステータスデコーダ313からの制御信号を受信する。フロー解析ユニット312はプログラムの実行前に予め解析したフローデータたとえばブロックの最初のアドレスを記憶している。また、フロー解析ユニット312は分岐成立/不成立信号S<sub>j</sub>及びラッチ回路314のアドレスを用いて新しいブロックのアドレスを発生し、これをトレースアドレスとしてトレースメモリ32に送出する。同時にラッチ回路314の内容はフロー解析ユニット312によって更新される。なお、ここで、“ブロック”とは、シーケンシャルに実行される命令の束である。言い換えると、1つのブロック内においては、分岐も合流もない。

【0028】トレース制御ユニット311及びフロー解析ユニット312の組合せの動作は図7、図8及び図9により明確に理解される。図7においては、分岐命令として分岐先（アドレス）が予め計算つまり決定できる静的分岐命令のみを含むものとする。図7を参照すると、ステップ901において、検出信号S<sub>i</sub>に従って分岐命令が実行されたか否かを検出する。分岐命令が実行された場合のみ、制御はステップ902に移行する。他の場合にはステップ901に戻る。

【0029】ステップ902では、ステータスデコーダ313の出力に従ってバスアクセスサイクルたとえば命令アクセスサイクル、キャッシュリプレイスサイクル等のバスアクセスサイクルが存在するか否かを判別する。この結果、バスアクセスサイクルが存在すれば、制御はステップ904に進み、通常のトレースを実行する。すなわち、命令アドレスを外部アドレスバスABから読み込み、トレースアドレスとしてトレースメモリ32に送出する。この場合、トレース制御ユニット311はストローブ信号SRを発生すると同時に、ラッチ回路314を動作させる。他の場合にはステップ903に進む。

【0030】ステップ903では、分岐成立/不成立信号S<sub>j</sub>に従って分岐が成立したか否かを判別する。分岐が不成立であればステップ905に進み、分岐が成立であれば、ステップ906に進む。

【0031】ステップ905では、次のブロックの最初

のアドレスのトレースを実行する。つまり、上述のごとく、フロー解析ユニット312は命令フローを前もって解析しているので、フロー解析ユニット312は前もって各ブロックの最初のアドレスを認識している。従って、この場合、フロー解析ユニット312が分岐は不成立であることを示す信号S<sub>j</sub>を受信すると、フロー解析ユニット312は次のブロックの最初のアドレスをトレースアドレスとしてラッチ回路314を介してトレースメモリ32に送出する。

【0032】ステップ906では、分岐先ブロックの最初のアドレスのトレースが実行される。すなわち、フロー解析ユニット312が分岐は成立であることを示す信号を受信すると、フロー解析ユニット312は分岐先（アドレス）を含む分岐先ブロックの最初のアドレスをトレースアドレスとしてトレースメモリ32に送出する。

【0033】ステップ904、905もしくは906での制御はステップ901に戻る。次に、分岐命令として分岐先（アドレス）を予め計算つまり決定することができない動的な分岐命令をも含むと仮定する。この場合、図1のマイクロプロセッサ1において、レジスタ間接命令のような動的な分岐命令が実行されると、制御は図8に示すトレーストラップルーチン（処理）へジャンプし、外部アドレスバスABに分岐先（アドレス）を送出する。この場合、トレース解析ユニット312は分岐先を解析して記憶している。従って、トレース制御ユニット311及びフロー解析ユニット312の組合せは図9のごとく動作する。

【0034】図9においては、ステップ1101、1102が図7に付加されている。つまり、ステップ903にて分岐が成立したと判別されたときはステップ1101に進み、ステータスデコーダ313の出力に従ってマイクロプロセッサ1においてトレーストラップ処理が実行されたか否かを判別する。トレーストラップ処理が実行されていないければ、ステップ906に進み、次のブロックの最初のアドレスをトレースする。逆に、トレーストラップ処理が実行されていれば、ステップ1102に進む。

【0035】ステップ1102では、フロー解析ユニット312はトレーストラップ処理において既に記憶されていた分岐先（アドレス）を発生し、これをトレースアドレスとしてラッチ回路314を介してトレースメモリ32に送出する。図9においては、ステップ903にて分岐が成立したと判別した後は、始めに分岐先ブロックの最初のアドレスをトレースした後に、トレーストラップ処理の実行を検出したならトレーストラップ処理によるアドレス出力のトレースに差し替えることもできる。

【0036】図10は本発明に係るマイクロプロセッサの第2の実施例を示す。図10においては、図1の構成



要素にステータスエンコーダ17を付加してある。このステータスエンコーダ17は検出信号S<sub>1</sub>及び分岐成立/不成立信号S<sub>2</sub>をステータス信号線STに送出するために用いるものである。この結果、図10のマイクロプロセッサ1に対してトレーサを構成する場合には、マイクロプロセッサ1とトレーサ3との間に特別な配線は不要となる。従って、トレーサシステムは図23に示すものと同様なものとして構成できる。

【0037】ステータスエンコーダ17は図11に示す4つのステータスビットST3、ST2、ST1、ST0を発生する論理ゲート回路によって構成することができる。また、トレーサアナライザ31は図12に示す回路によって構成できる。すなわち、図12に示すごとく、ステータスデコーダ313は4つのステータスビットST3、ST2、ST1及びST0をデコードしてデコード信号をトレーサ制御ユニット311、フロー解析ユニット312及びラッチ回路314に送出する。この場合、図12のトレーサ制御ユニット311及びフロー解析ユニット312の組合せは図6の場合と同様に動作し、従って、図7、図9のフローチャートで示すごとく動作する。

【0038】図13は本発明に係るマイクロプロセッサの第3の実施例を示す。図13においては、図1の構成要素にキャッシュ制御レジスタ18を付加してある。このキャッシュ制御レジスタ18は内部データバスIDBに接続されており、これにより、命令キャッシュ12が活性化しているか否かを判別する。すなわち、キャッシュ制御レジスタ18はキャッシュオン/オフ信号S<sub>3</sub>を発生し、これを外部へ送出すると共に命令キャッシュ12に送出する。

【0039】キャッシュ制御レジスタ18が命令キャッシュ12を活性化するキャッシュオン信号S<sub>3</sub>を発生したときには、命令キャッシュ12は活性化される。この場合、命令キャッシュ12がヒットすると、命令キャッシュ12は命令コードを命令デコーダ11に送出する。逆に、命令キャッシュ12がミスヒットすると、命令キャッシュ12はフェッチ要求をミスヒットした命令アドレスと共にバス制御ユニット14に送出する。

【0040】他方、キャッシュ制御レジスタ18が命令キャッシュ12を非活性化にするキャッシュオフ信号S<sub>3</sub>を送出すると、命令キャッシュ12は非活性化となる。この場合、命令キャッシュ12がヒットしても、命令キャッシュ12はヒットした命令アドレスと共にフェッチ要求をバス制御ユニット14に送出する。

【0041】図14は図13の命令キャッシュ12の詳細なブロック回路図である。図14においては、キャッシュ制御レジスタ18のキャッシュオン/オフ信号S<sub>3</sub>がキャッシュ制御ユニット124に供給されている。キャッシュ制御レジスタ18が命令キャッシュ12を活性化するキャッシュオン信号S<sub>3</sub>を発生したときには、キ

ャッシュ制御ユニット124は図2の場合と同様に動作する。逆に、キャッシュ制御レジスタ18が命令キャッシュ12を非活性化にするキャッシュオフ信号S<sub>3</sub>を発生したときには、比較器123のヒット/ミスヒット出力に関係なく、キャッシュ制御ユニット124はラッチ回路126からの命令アドレスと共にフェッチ要求を発生する。また、この場合、キャッシュ制御ユニット124はセレクタ125を制御して内部データバスIDBを選択させる。

【0042】図15は図13のマイクロプロセッサ1のトレーサシステムを示す。図15においては、マイクロプロセッサ1のキャッシュオン/オフ信号S<sub>3</sub>がトレーサアナライザ31に供給されている点が、図4の場合と異なる。この結果、トレーサアナライザ31においては、図16に示すごとく、図6の構成要素に対してアンド回路315、316が付加されている。すなわち、キャッシュオン/オフ信号S<sub>3</sub>が“1”（キャッシュオン状態）を示しているときには、アンド回路315、316はイネーブルとなって2つの信号S<sub>1</sub>、S<sub>2</sub>を通過させ、この結果、トレーサアナライザ31は図6の場合と同様に動作する。逆に、キャッシュオン/オフ信号S<sub>3</sub>が“0”（キャッシュオフ状態）を示しているときには、アンド回路315、316はディセーブルとなり、従って、トレーサ制御ユニット311及びフロー解析ユニット312の組合せは通常のトレーサを行う。つまり、この場合、図13のマイクロプロセッサ1はキャッシュなしのマイクロプロセッサとして動作し、従って、すべての命令アクセスはマイクロプロセッサ1の外部に送出されるからである。

【0043】分岐命令は静的分岐命令のみを含むものとすれば、図16のトレーサ制御ユニット311、フロー解析ユニット312及びアンド回路315、316の組合せ図17に示すごとく動作する。図17においては、ステップ1901が図7に付加されている。すなわち、ステップ1901では、キャッシュオン/オフ信号S<sub>3</sub>に従って命令キャッシュ12が活性化（オン状態）か否かを判別する。この結果、命令キャッシュ12が非活性化（オフ）されていれば、ステップ904に進み、通常のトレーサを行う。逆に、命令キャッシュ12が活性化（オン）されていれば、ステップ901に進み、その後は図7に示す動作が実行されることになる。

【0044】また、分岐命令は静的分岐命令及び動的分岐命令を含むものとすれば、図16のトレーサ制御ユニット311、フロー解析ユニット312及びアンド回路315、316の組合せ図18に示すごとく動作する。図18においては、ステップ2001が図9に付加されている。すなわち、ステップ2001では、キャッシュオン/オフ信号S<sub>3</sub>に従って命令キャッシュ12が活性化（オン状態）か否かを判別する。この結果、命令キャッシュ12が非活性化（オフ）されていれば、ステップ



904に進み、通常のトレースを行う。逆に、命令キャッシュ12が活性化(オン)されていれば、ステップ901に進み、その後は図9に示す動作が実行されることになる。

【0045】図19は本発明に係るマイクロプロセッサの第4の実施例を示す。図19においては、図13の構成要素にステータスエンコーダ17'を付加してある。このステータスエンコーダ17'は検出信号S<sub>1</sub>、分岐成立/不成立信号S<sub>2</sub>、キャッシュオン/オフ信号S<sub>3</sub>をステータス信号線STに送出するために用いるものである。この結果、図13のマイクロプロセッサ1に対してトレースを構成する場合には、マイクロプロセッサ1とトレース3との間に特別の配線は不要となる。従って、トレースシステムは図23に示すものと同等なものとして構成できる。

【0046】ステータスエンコーダ17'は図20に示す5つのステータスビットST4、ST3、ST2、ST1、ST0を発生する論理ゲート回路によって構成することができる。また、トレースアナライザ31は図21に示す回路によって構成できる。すなわち、図21に示すごとく、ステータスデコーダ313'は5つのステータスビットST4、ST3、ST2、ST1及びST0をデコードしてデコード信号をトレース制御ユニット311、フロー解析ユニット312及びラッチ回路314に送出する。この場合、図21のトレース制御ユニット311及びフロー解析ユニット312の組合せは図14の場合と同様に動作し、従って、図17、図18のフローチャートで示すごとく動作する。

【0047】なお、上述の実施例においては、動的分岐命令としてレジスタ間分岐命令を例としたが、本発明は例外処理の実行による例外処理プログラムにも適用できる。また、図7、図9、図17及び図18においては、ブロックの最初のアドレスをトレースしているが、命令フローの解析に役立つブロック番号のようなブロックを代表する番号をトレースしてもよい。

【0048】

【発明の効果】以上説明したように本発明によれば、分岐命令の実行に関する情報及び分岐が成立したか否かの情報をキャッシュ内蔵マイクロプロセッサの外部に出力しているので、たとえ命令キャッシュがヒットしても、これらの情報を用いて命令のトレースを実行して効率的なデバッグを可能にしている。

【図面の簡単な説明】

【図1】本発明に係るマイクロプロセッサの第1の実施例を示すブロック回路図である。

【図2】図1の命令キャッシュの詳細なブロック回路図である。

【図3】図1のバス制御ユニットの詳細なブロック回路図である。

【図4】図1のマイクロプロセッサに用いられる命令コ

ードの一例を示す。

【図5】図1のマイクロプロセッサのためのトレースシステムを示すブロック回路図である。

【図6】図5のトレースアナライザの詳細なブロック回路図である。

【図7】図5のトレースの動作を示すフローチャートである。

【図8】図1のマイクロプロセッサにおいて動作するトレーストラップ処理を示す図である。

【図9】図5のトレースの動作を示すフローチャートである。

【図10】本発明に係るマイクロプロセッサの第2の実施例を示すブロック回路図である。

【図11】図10のステータスエンコーダの出力を示す図である。

【図12】図10のマイクロプロセッサ用トレースアナライザの詳細なブロック回路図である。

【図13】本発明に係るマイクロプロセッサの第3の実施例を示すブロック回路図である。

【図14】図13の命令キャッシュの詳細なブロック回路図である。

【図15】図13のマイクロプロセッサのためのトレースシステムを示すブロック回路図である。

【図16】図15のトレースアナライザの詳細なブロック回路図である。

【図17】図15のトレースの動作を示すフローチャートである。

【図18】図15のトレースの動作を示すフローチャートである。

【図19】本発明に係るマイクロプロセッサの第4の実施例を示すブロック回路図である。

【図20】図19のステータスエンコーダの出力を示す図である。

【図21】図19のマイクロプロセッサのためのトレースアナライザの詳細なブロック回路図である。

【図22】従来のマイクロプロセッサを示すブロック回路図である。

【図23】図22のマイクロプロセッサのためのトレースシステムを示すブロック回路図である。

【符号の説明】

1…マイクロプロセッサ

2…外部メモリ

3…トレース

11…命令デコーダ

12…命令キャッシュ

13…実行ユニット

14…バス制御ユニット

15…オア回路

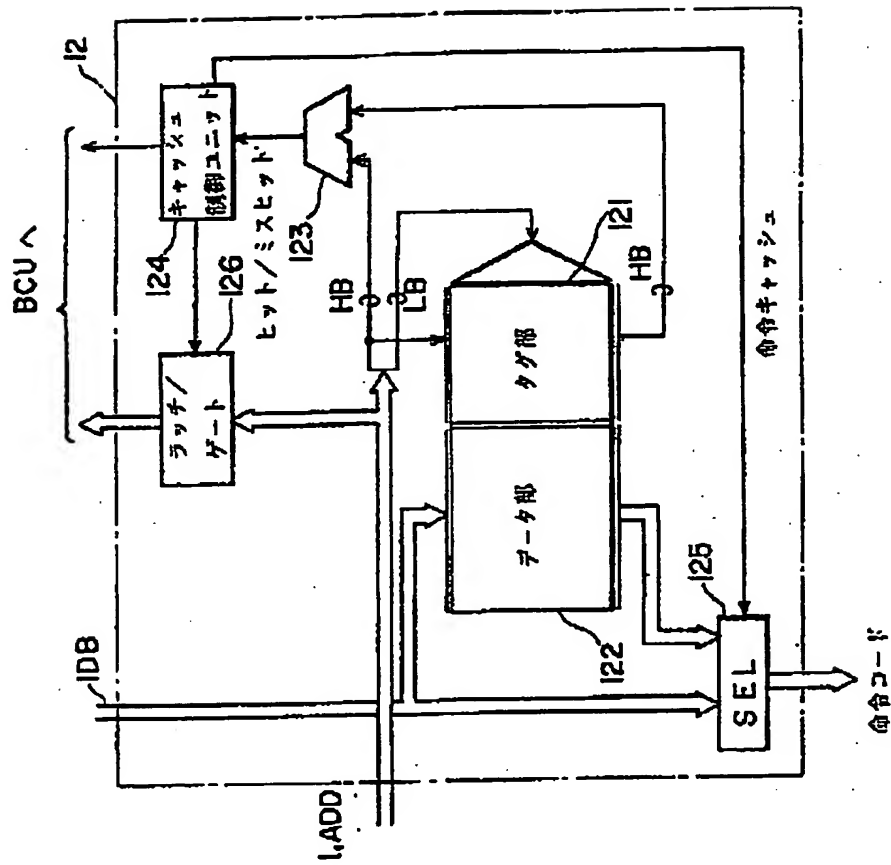
16…割り込み検出ユニット

17、17'…ステータスエンコーダ



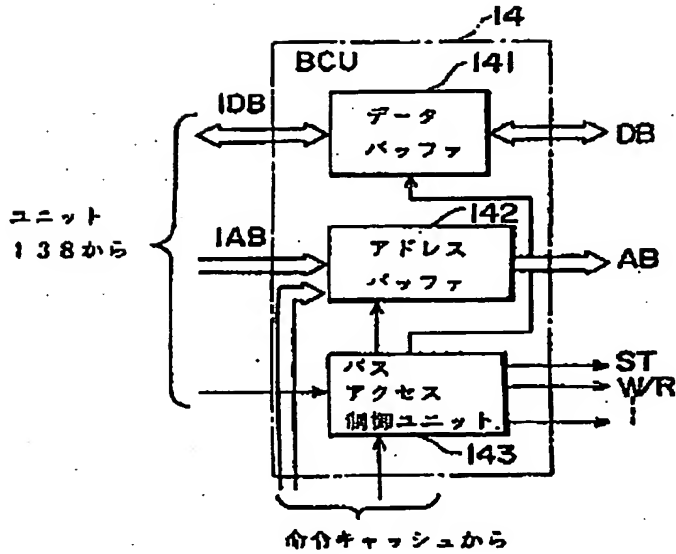
【図2】

図1の命令キャッシュ



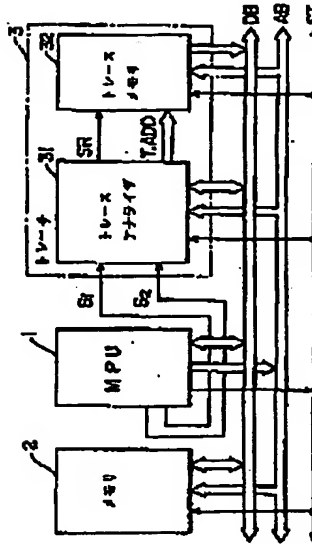
【図3】

図1のバス制御ユニット



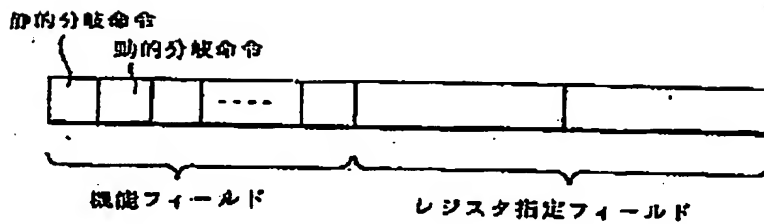
【図5】

図1のマイクロプロセッサ用トレースシステム



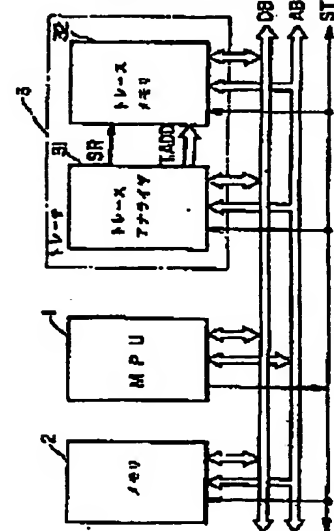
【図4】

命令コードの一例



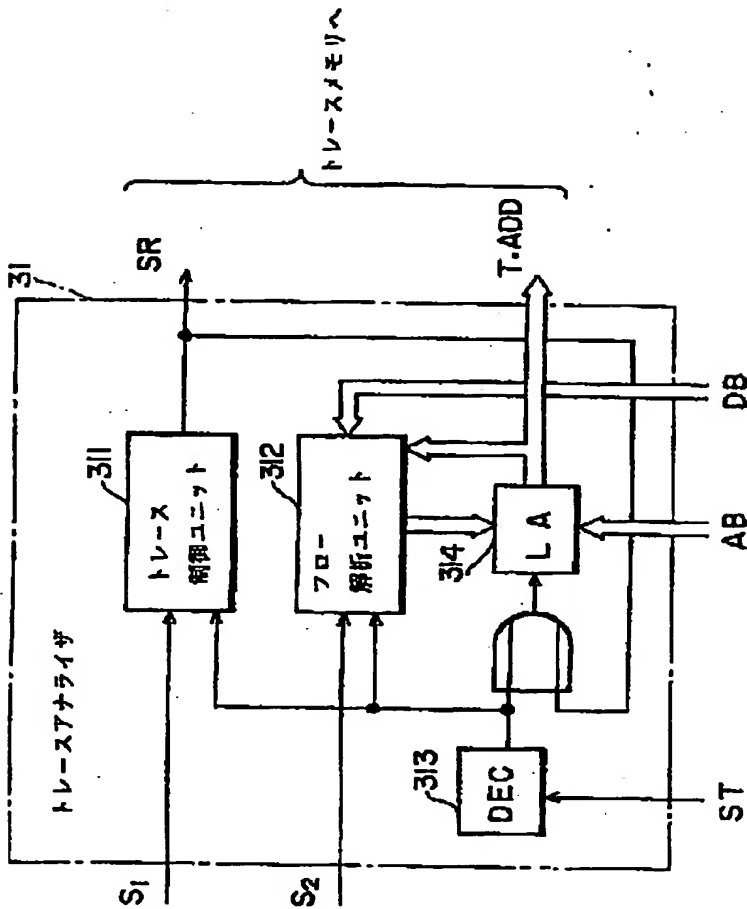
【図23】

従来のトレースシステム



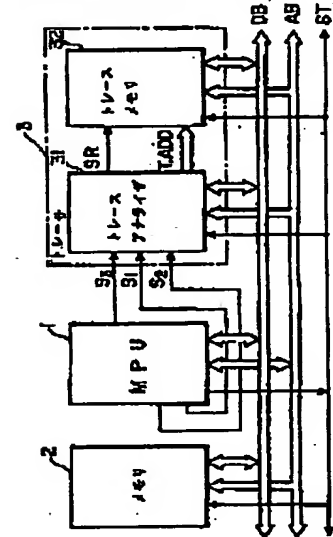
【図6】

図5のトレースアナライザ



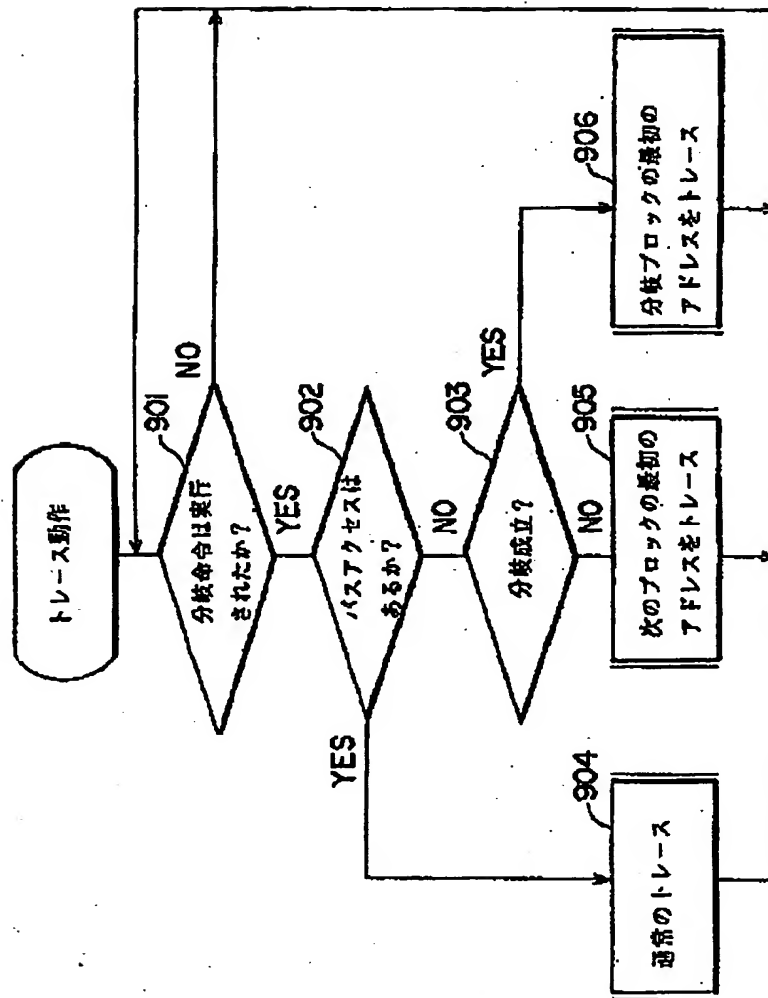
【図15】

図13のマイクロプロセッサ用トレースシステム



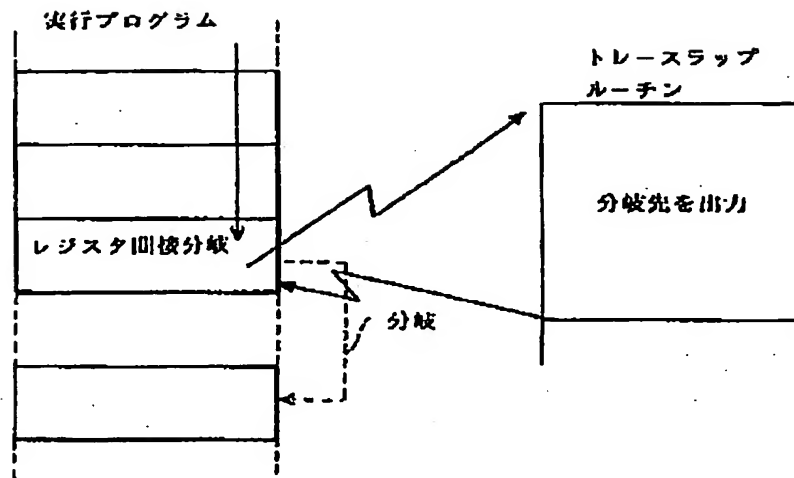
【図7】

図6の動作



【図8】

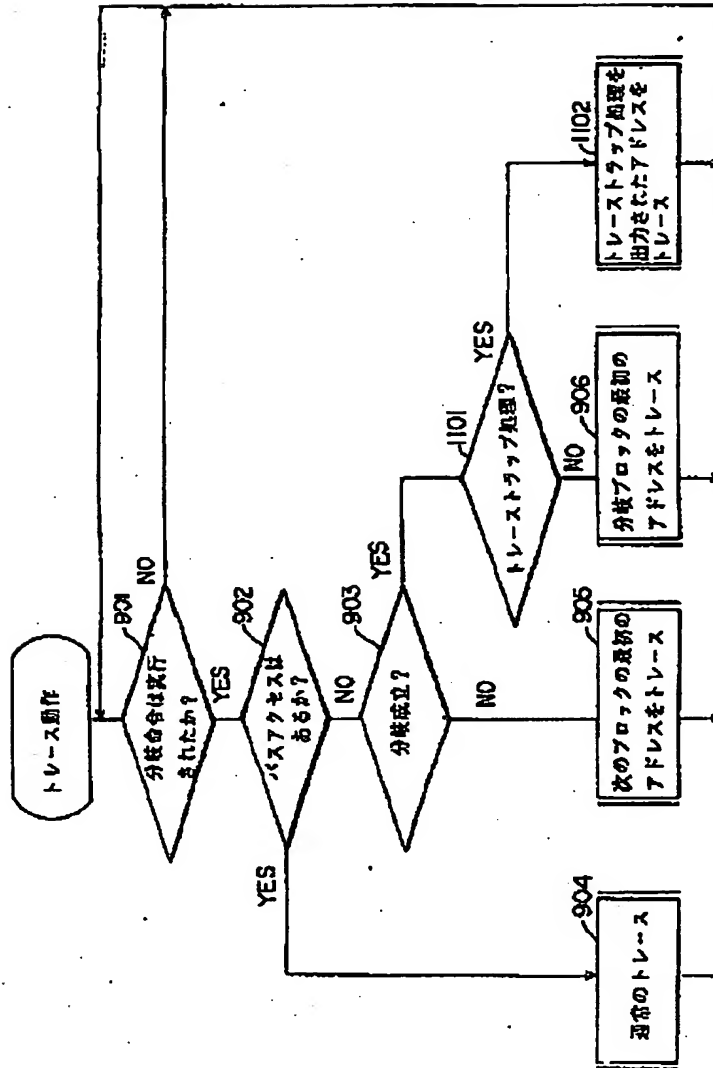
図1のマイクロプロセッサにおける  
トレースラップ処理





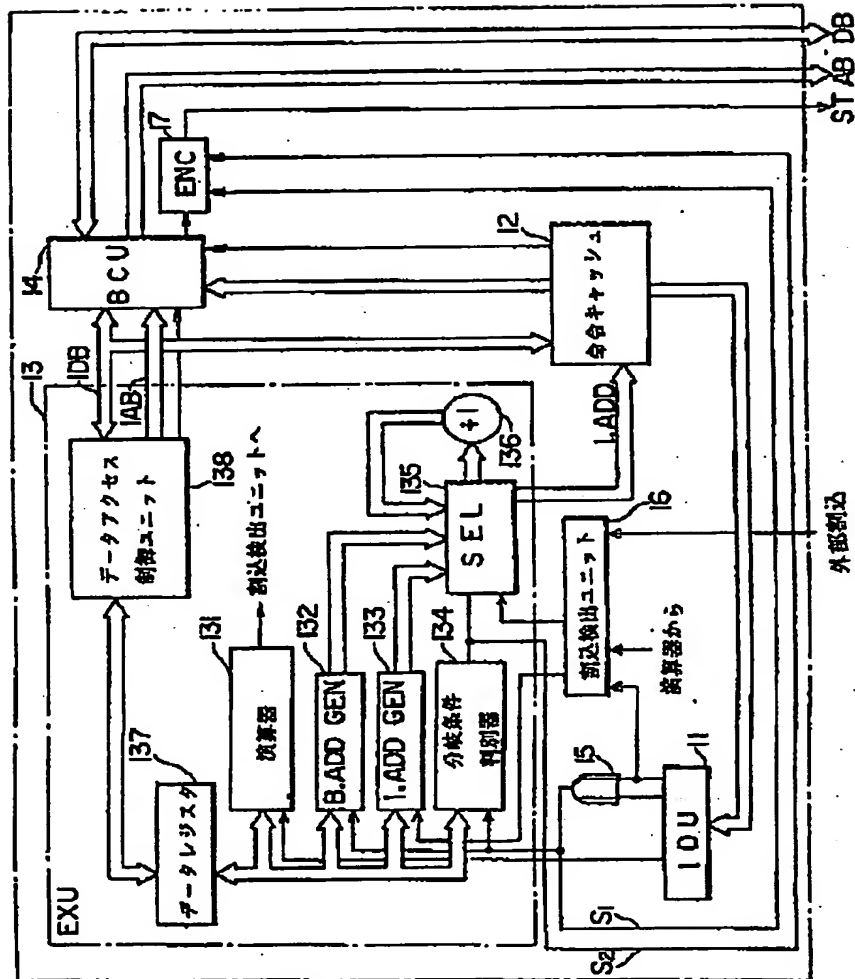
【図9】

図6の動作



【図10】

本発明の第2の実施例



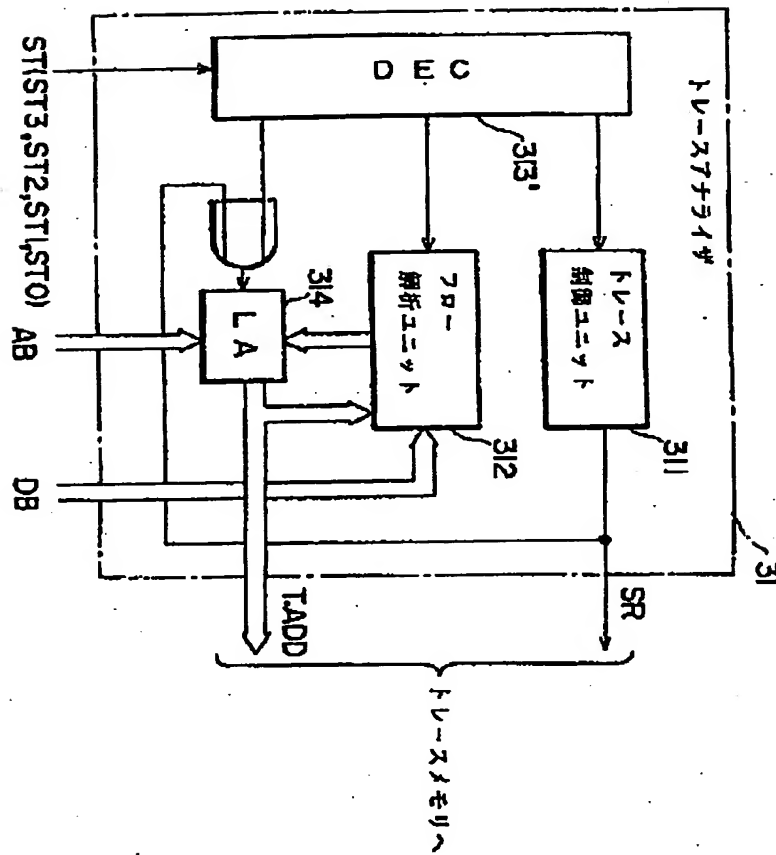
【図11】

図10のステータスエンコードの論理構成

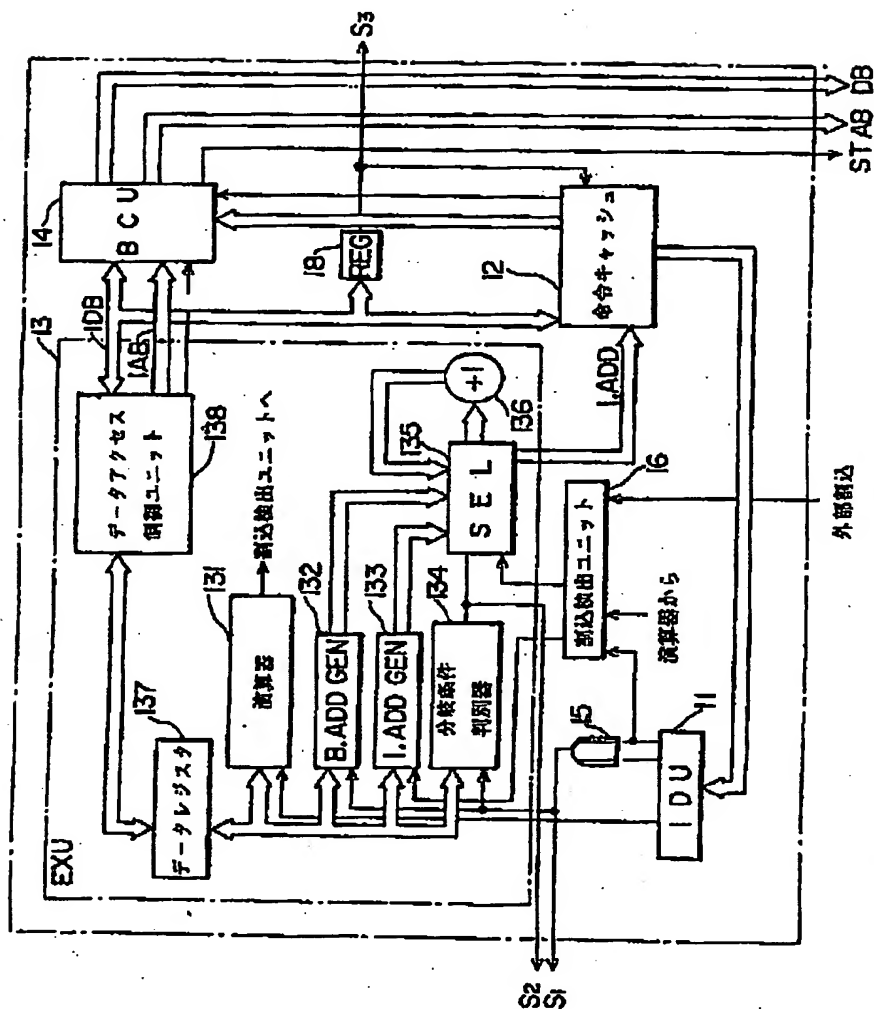
	ST	ST3	ST2	ST1	ST0
非トレーストラップ	データ アクセス	0	0	0	0
	命令 アクセス	0	0	0	1
	キャッシュ リプレイス	0	0	1	0
	分岐 (不成立)	0	0	0	0
	分岐 (成立)	0	0	0	1
トレーストラップ	データ アクセス	1	0	0	0
	命令 アクセス	1	0	0	1
	キャッシュ リプレイス	1	0	0	0
	分岐 (不成立)	1	1	0	0
	分岐 (成立)	1	1	0	1

【図12】

図10に対するトレースアナライザ

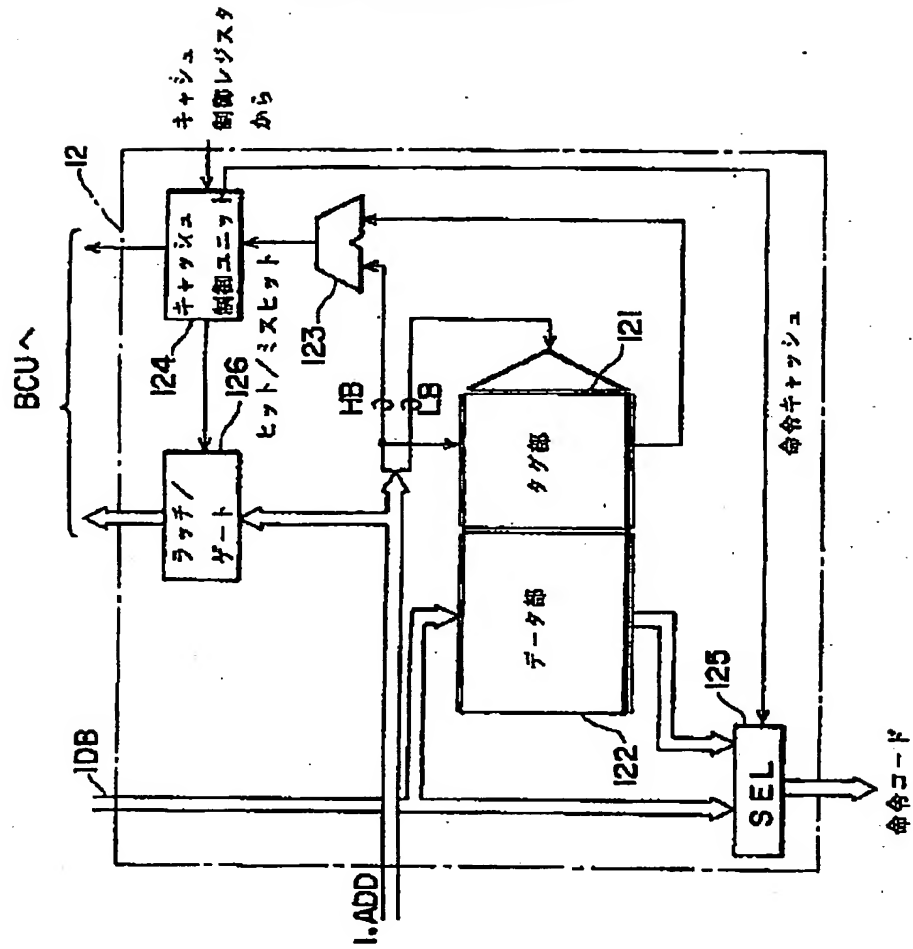


### 木発明の第3の実施例



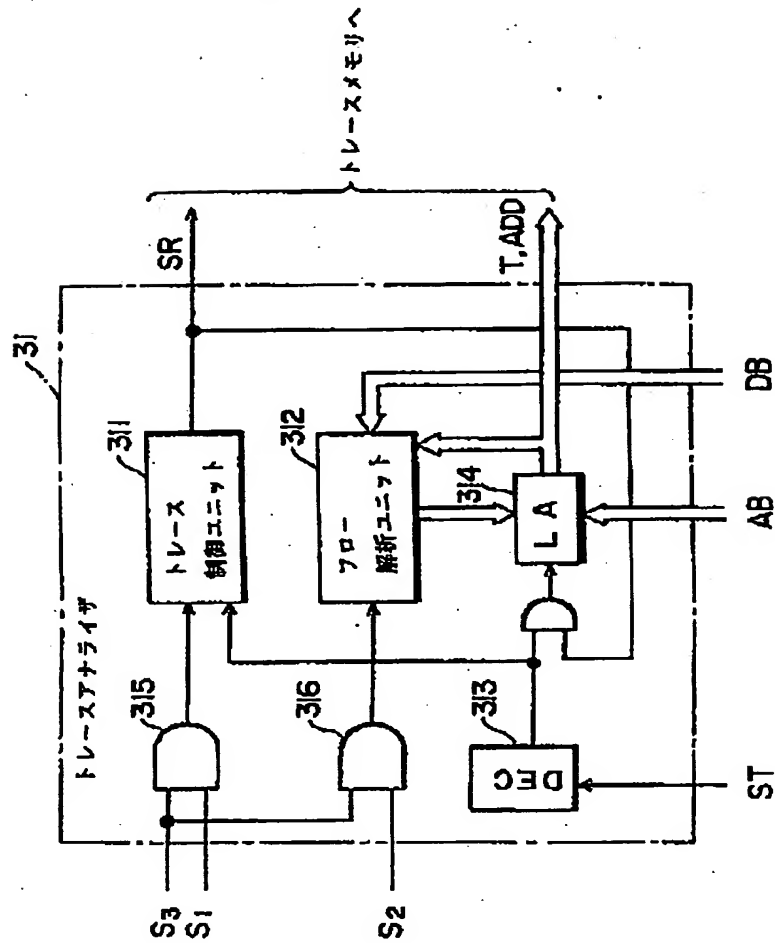
【図14】

図13の命令キャッシュ



【図16】

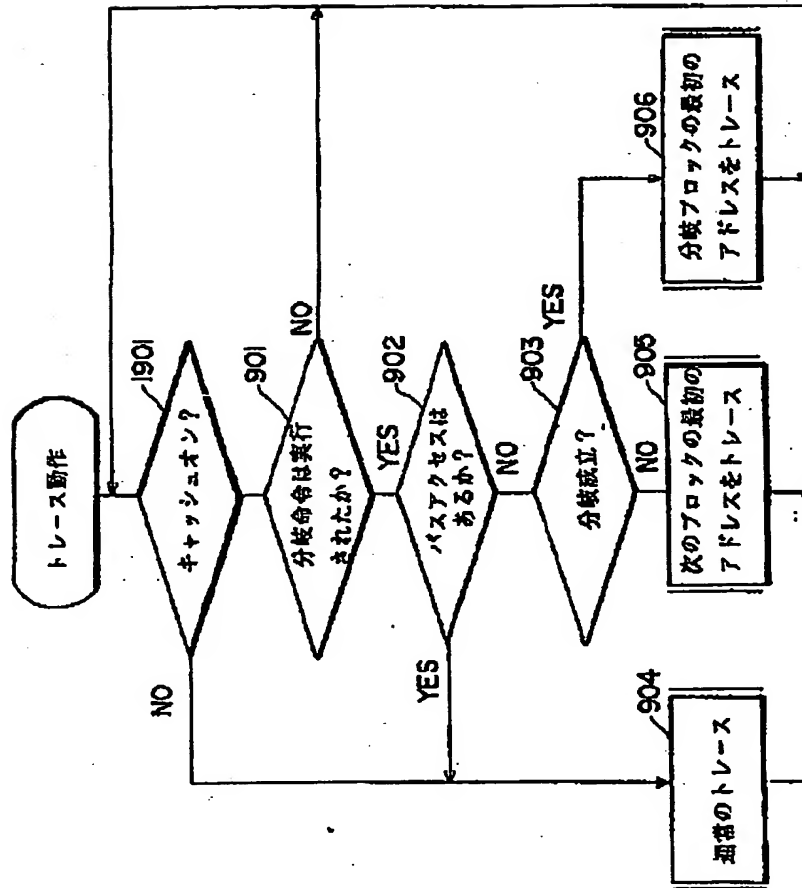
図15のトレースアナライザ





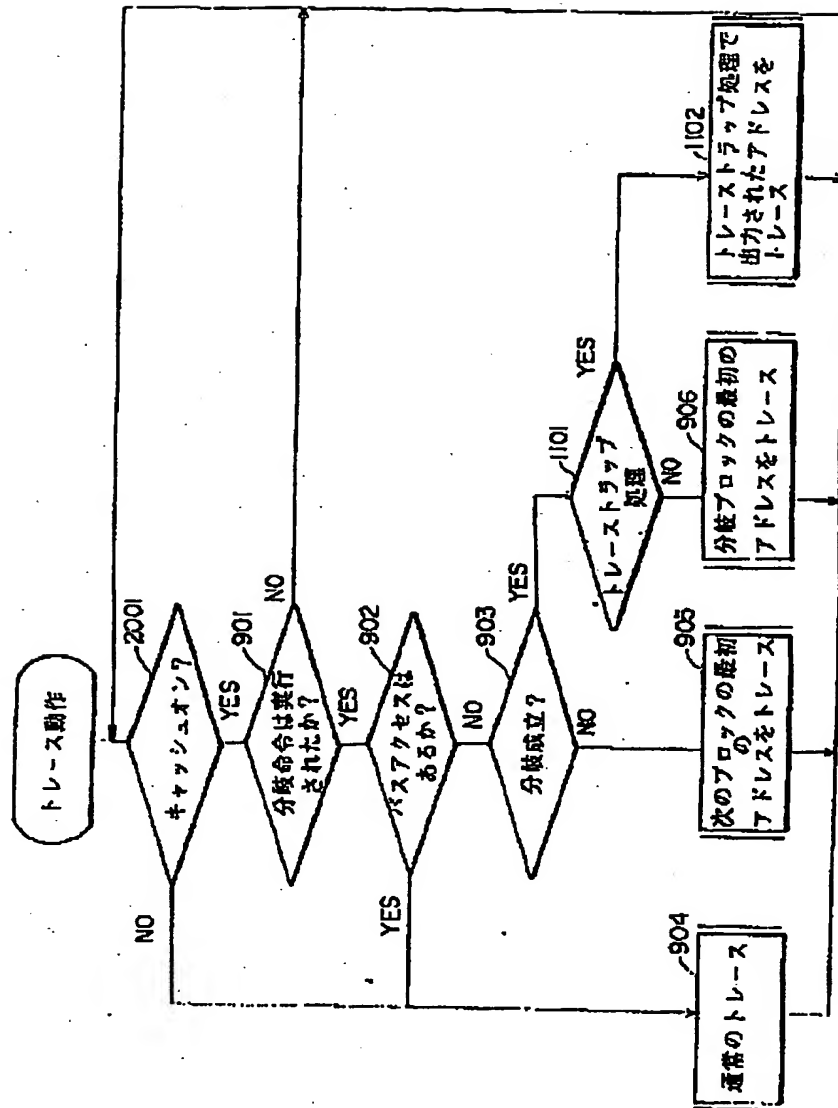
【図17】

図16の動作

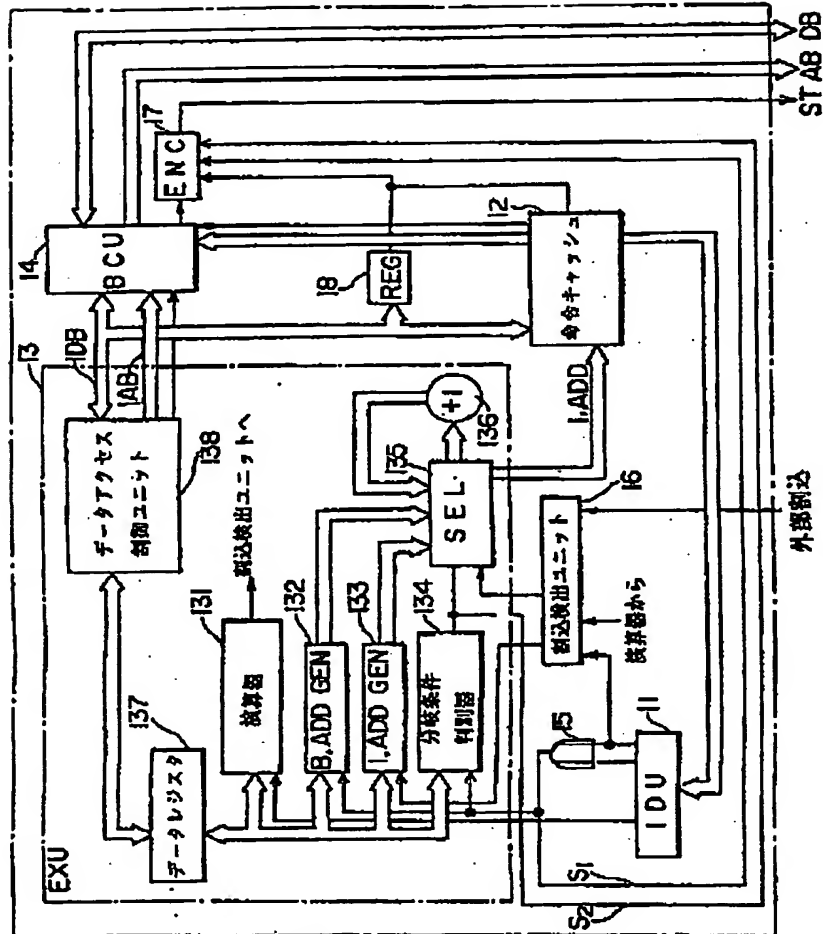


【図18】

図16の動作



### 木彫りの第1の奨励例



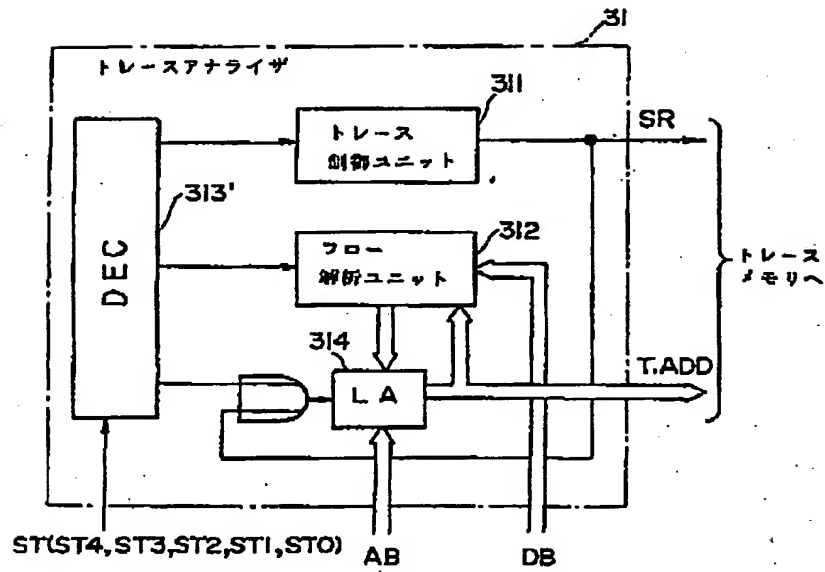
【図20】

図19のステータスエンコードの論理構成

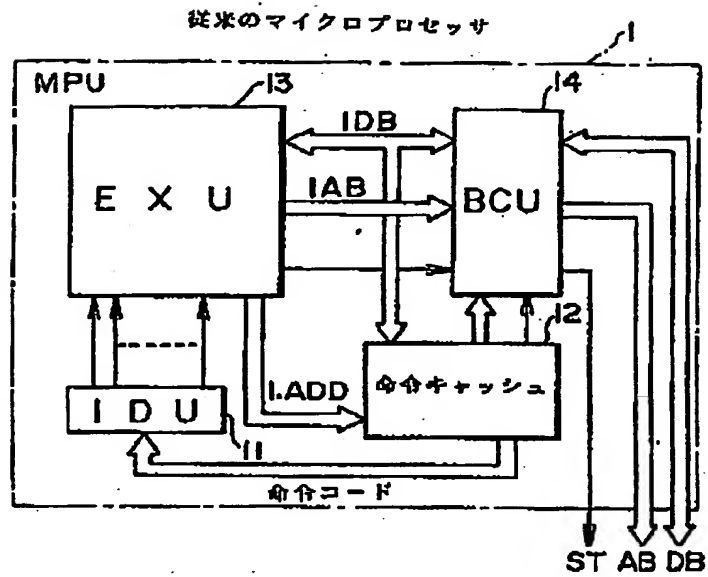
	ST	ST4	ST3	ST2	ST1	ST0
非トレーストラップ	データ アクセス	0	0	0	0	0
	命令 アクセス	0	0	0	0	1
	キャッシュ リプレース	0	0	0	1	0
	分岐 (不成立)	0	0	1	0	0
	分岐 (成立)	0	0	1	0	1
	キャッシュオン	0	1	0	0	0
	キャッシュオフ	0	1	0	0	1
トレーストラップ	データ アクセス	1	0	0	0	0
	命令 アクセス	1	0	0	0	1
	キャッシュ リプレース	1	0	0	1	0
	分岐 (不成立)	1	0	1	0	0
	分岐 (成立)	1	0	1	0	1
	キャッシュオン	1	1	0	0	0
	キャッシュオフ	1	1	0	0	1

【図21】

図19に対するトレースアナライザ



**【例 2 2】**



# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-075858  
(43)Date of publication of application : 18.03.1994

(51)Int.Cl.

G06F 12/08  
G06F 11/28  
// G06F 9/32

(21)Application number : 05-128091  
(22)Date of filing : 30.04.1993

(71)Applicant : NEC CORP  
(72)Inventor : ARAI TOMOHISA  
HORIGUCHI YUMIKO

(30)Priority

Priority number : 04119403    Priority date : 13.05.1992    Priority country : JP  
04187497    22.06.1992

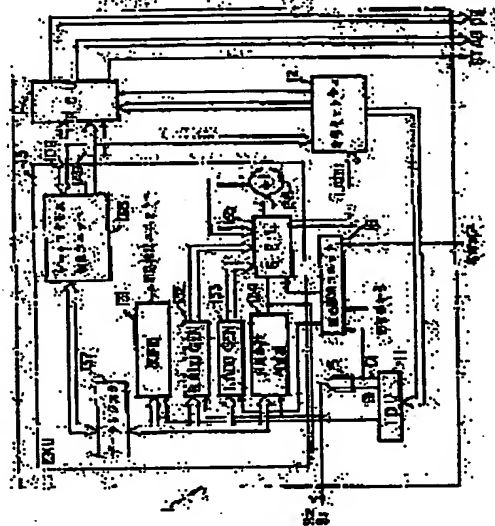
JP

## (54) CACHE BUILT-IN MICROPROCESSOR AND ITS TRACE SYSTEM

(57)Abstract:

PURPOSE: To trace the execution of an instruction of a microprocessor at the time of hit by informing the outside of a branch instruction and formation/ unformation of its branch instruction, when the branch instruction is executed.

CONSTITUTION: An OR circuit 15 generates a detecting signal S1 for detecting a branch instruction to be executed and outputs it directly to the outside of a microprocessor 1. When the branch is formed, a selector 135 selects a branch address and send it out as an instruction address to an instruction cache 12. Also, when an instruction is generated, the selector 135 selects an interruption address and sends it out as an instruction address to the instruction cache 12. On the contrary, when the branch is unformed or the interruption is not generated, the selector 135 selects an output of a +1 incrementer 136 and sends it out to the instruction cache 12. Moreover, as a result of execution of the branch instruction, a branch condition discriminator 134 generates a branch formation/unformation signal S2 for showing whether the branch is formed or not and sends it out directly to the outside of the microprocessor 1.



## LEGAL STATUS

[Date of request for examination] 30.04.1993

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 2646957

[Date of registration] 09.05.1997



[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

09.05.2003

Copyright (C); 1998,2003 Japan Patent Office

\* NOTICES \*

- 5 JPO and NCIP are not responsible for any damages caused by the use of this translation.  
1.This document has been translated by computer. So the translation may not reflect the original precisely.  
2.\*\*\*\* shows the word which can not be translated.  
3.In the drawings, any words are not translated.

10

CLAIMS

- 15 [Claim(s)]  
[Claim 1] The microprocessor with a built-in cache provided in a branch instruction detection means detect the branch instruction which should be executed, a notice means of branch instruction detection notify outside whether said branch instruction was detected, the branching formation / abortive distinction means that branching was materialized by  
20 activation of said branch instruction, or distinguishes no, and the branching formation / abortive notice means which notifies outside whether said branching was materialized.  
[Claim 2] Said each notice means is a microprocessor according to claim 1 with a built-in cache which uses an external status signal terminal.  
[Claim 3] Furthermore, the microprocessor according to claim 1 with a built-in cache provide a  
25 dynamic branch-instruction distinction means distinguish whether said branch instruction is the dynamic branch instruction which cannot determine a branching place statically beforehand, an interrupt generating means generate interruption when said dynamic branch instruction is detected, and a trace trap processing means performs trace trap processing, calculates said branching place, and output outside when this interrupt occurs.  
30 [Claim 4] Furthermore, the microprocessor possessing an instruction-cache activity distinction means to distinguish whether the instruction cache (12) is activated, and the cache activity / notice means of non-activity which notifies outside whether said instruction cache is being activated according to claim 1 with a built-in cache.  
[Claim 5] Said each notice means is a microprocessor according to claim 4 with a built-in cache which uses an external status signal terminal (ST).  
35 [Claim 6] It connects with the EU (13) which generates the instruction address, and this EU. When said instruction address is received and this instruction address is registered, the instruction code corresponding to this instruction address is generated. The instruction cache which generates a replacement fetch demand to the exterior when said instruction address is  
40 not registered (12), The instruction decoder which generates the control signal which is connected to this instruction cache and said EU, decodes the instruction code from said

instruction cache, and includes a decision instruction in said EU (11), A branch instruction generating output means to be connected to said instruction decoder and to output generating of said branch instruction outside. The microprocessor possessing a branch instruction formation output means to output formation of this branching to the exterior when it connects with said RU, this EU executes said branch instruction and branching is materialized.

5 [Claim 7] Furthermore, it is the microprocessor according to claim 6 for which it connects with said EU and said instruction cache, the bus control unit (14) which outputs the status signal which shows the condition of said microprocessor to the exterior is provided, and said branch instruction generating output means and said branch instruction formation output means use

10 the status signal terminal of said microprocessor.

[Claim 8] Furthermore, it is the microprocessor according to claim 6 which possesses an output means for said EU to receive said interruption, to perform trace trap processing, and to output said branching place to the exterior, by providing an interrupt generating means to generate interruption at the time of the dynamic branch instruction as which said branch instruction

15 cannot determine a branching place statically beforehand.

[Claim 9] Furthermore, the microprocessor possessing the cache control register (18) which is connected to said EU and said instruction cache, activates / deactivates this instruction cache, and outputs cache-on / OFF signal to the exterior according to the activity / non-activity of this instruction cache according to claim 7.

20 [Claim 10] Output the access signal accompanied by the instruction address which requires access to the branching formation / non-materialized signal (S2), and external memory (2) which show whether branching was materialized when the detecting signal (S1) and this branch instruction which show whether the branch instruction which should be executed was detected were executed. A means to be the trace system of a microprocessor (1) with

25 instruction-cache (13) built-in, and to distinguish whether said detecting signal was received and branch instruction was executed, A means to distinguish whether said branching formation / non-materialized signal were received, and branching was materialized, When a means to distinguish whether said access signal was received and it was accessed by said external memory, and branch instruction are executed and it is accessed by said external

30 memory When branch instruction is executed, and branching after activation of said branch instruction carries out failure to a means to trace said outputted instruction address and it is not accessed by said external memory The trace system which possesses a means to trace the block of a branching place when a means to trace the following block, and branch instruction are executed, and branching after activation of said branch instruction is materialized and it is

35 not accessed by said external memory.

[Claim 11] The access signal accompanied by the instruction address which requires access is outputted to the branching formation / abortive signal (S2), and external memory (2) which show whether branching was materialized when the detecting signal (S1) and this branch instruction which show whether the branch instruction which should be executed was detected

40 were executed. When said branch instruction is the dynamic branch instruction for which it cannot opt statically beforehand, perform trace trap processing and output the branching place

address to the exterior. A means to be the trace system of a microprocessor (1) with  
 instruction-cache (13) built-in, and to distinguish whether said detecting signal was received  
 and branch instruction was executed, A means to distinguish whether said branching  
 formation / abortive signal were received, and branching was materialized, When a means to  
 5 distinguish whether said access signal was received and it was accessed by said external  
 memory, and branch instruction are executed and it is accessed by said external memory When  
 branch instruction is executed, and branching after activation of said branch instruction  
 carries out failure to a means to trace said outputted instruction address and it is not accessed  
 by said external memory A means to trace the following block, and branch instruction are  
 10 executed, and branching after activation of said branch instruction is materialized. A means to  
 trace the block of a branching place when it is not accessed by said external memory and trace  
 trap processing is not performed, The trace system possessing a means to trace the branching  
 place address outputted by this trace trap processing when branch instruction is executed,  
 branching after activation of said branch instruction is materialized, and it is not accessed by  
 15 said external memory and trace trap processing is performed.  
 [Claim 12] The branching formation / abortive signal which shows whether branching was  
 materialized when the detecting signal (S1) and this branch instruction which show whether  
 the branch instruction which should be executed was detected were executed (S2), Output the  
 cache-on / off signal (S3) which activates / deactivates the access signal and instruction cache  
 20 (13) accompanied by the instruction address which requires access to external memory (2). A  
 means to be the trace system of a microprocessor (1) with a built-in instruction cache, and to  
 distinguish whether said detecting signal was received and branch instruction was executed, A  
 means to distinguish whether said branching formation / abortive signal were received, and  
 branching was materialized, A means to distinguish whether said access signal was received  
 25 and it was accessed by said external memory, When a means or or to distinguish [ to which  
 said cache on-off signal is received and said instruction cache is being activated ] whether  
 deactivation is carried out, and said instruction cache are deactivated, Or when branch  
 instruction is executed and it is accessed by said external memory When said instruction cache  
 is activated, branch instruction is executed, and branching after activation of said branch  
 30 instruction carries out failure to a means to trace said outputted instruction address and it is  
 not accessed by said external memory The trace system which possesses a means to trace the  
 block of a branching place when a means to trace the following block, and said instruction  
 cache are activated, branch instruction is executed, and branching after activation of said  
 branch instruction is materialized and it is not accessed by said external memory.  
 35 [Claim 13] The branching formation / abortive signal which shows whether branching was  
 materialized when the detecting signal (S1) and this branch instruction which show whether  
 the branch instruction which should be executed was detected were executed (S2), The cache  
 on-off signal (S3) which activates / deactivates the access signal and instruction cache (13)  
 accompanied by the instruction address which requires access is outputted to external memory  
 40 (2). When said branch instruction is the dynamic branch instruction for which it cannot opt  
 statically beforehand, perform trace trap processing and output the branching place address to

the exterior. A means to be the trace system of a microprocessor (1) with a built-in instruction cache, and to distinguish whether said detecting signal was received and branch instruction was executed, A means to distinguish whether said branching formation / abortive signal were received, and branching was materialized, A means to distinguish whether said access signal was received and it was accessed by said external memory, A means or or to distinguish [ which receives said cache on-off signal / by which said instruction cache is activated ] whether deactivation is carried out, and when said instruction cache is deactivated, Or when branch instruction is executed and it is accessed by said external memory When said instruction cache is activated, branch instruction is executed, and branching after activation of said branch instruction carries out failure to a means to trace said outputted instruction address and it is not accessed by said external memory When a means to trace the following block, and said instruction cache are activated, branch instruction is executed, branching after activation of said branch instruction is materialized, and it is not accessed by said external memory and trace trap processing is not performed A means to trace the block of a branching place, and said instruction cache are activated. The trace system possessing a means to trace the branching place address outputted by this trace trap processing when branch instruction is executed, branching after activation of said branch instruction is materialized, and it is not accessed by said external memory and trace trap processing is performed.

20

---

#### DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

25 [0001]

[Industrial Application] This invention relates to the microprocessor which builds in the cache which attained easy-ization of debugging in a system or program development, and its trace system.

[0002]

30 [Description of the Prior Art] In the system of a microprocessor, or development of a program, there is the approach of three level, trace, a trap, and a single step, as the debugging approach.

[0003] (1) Collect the information which shows the trace approach program execution location, and recognize assembly program execution sequence for these. Or the information about access of a memory operand is collected and the access sequence of an assembly memory operand is recognized for these. Generally in a bus cycle, the address signal, data signal, and status signal of a microprocessor which can be observed with an external terminal can constitute such information. Thus, a tracing facility can be realized, without adding a function special to a microprocessor, since above-mentioned information can be observed in the exterior of a microprocessor. Moreover, since it is not necessary to generate the interruption or the exception which interrupts the program execution which should be debugged, or since it is not necessary to keep a special bus cycle waiting to the program execution which should be

40

debugged, the timing of operation at the time of debugging is the same as that of the timing of operation at the time of un-debugging. However, since this trace approach detects a specific condition and does not interrupt program execution, it is the passive approach of only observing the condition of the program which should be debugged.

5 [0004] (2) the trap approach -- oh, if access of instruction code or a memory operand occurs in the location (address) by which Ecklonia specification was carried out, control will shift to the debug program called a debugger, and thereby still more detailed debugging will be performed. Unlike the trace approach, control shifts to a debugger only to required timing. Therefore, the hardware device called trap interrupt is needed and interruption/exception is correctly  
10 generated just before activation of the instruction code which has the address which this set up beforehand, or in immediately after. Consequently, the trap approach is unrealizable in a usual state. Moreover, the C case of a program cannot be known correctly. Furthermore, program execution cannot be interrupted, when it overruns recklessly, for example before the program started the trap if control does not start the trap defined beforehand.

15 [0005] (3) Whenever it executes single step approach 1 instruction, shift control to a debugger, from this, display the internal state of a microprocessor, for example, the condition of a general-purpose register, a processor status word (PSW), a program counter (PC), etc., and change a part of these conditions. Thus, since the internal state of a microprocessor is recognized for every activation of one instruction, the procedure of a program can be recognized  
20 correctly.

[0006] Thus, in the single step approach, whenever a microprocessor executes one instruction, control is shifted to a debugger. Therefore, the hardware device called single step interrupt is needed, and this generates interruption/exception for every activation timing of one instruction. Consequently, a single step cannot always be realized. furthermore -- since control is shifted to  
25 TEBAGGA for every activation of each instruction -- timing like a \*\* time-sharing lump routine with low \*\* program execution effectiveness -- debugging of critical processing completely differs from the impossible interior-action timing's at time of \*\* debugging actuation timing at the time of un-debugging.

[0007] Since the three debugging approaches have an advantage and a fault like \*\*\*, generally the combination of these debugging approaches is used. For example, the actual sequence of an instruction is first recognized by the trace approach. When control comes [ 2nd ]  
30 near the location (address) which has generated the problem, control is shifted to a debugger by the trap approach. Finally, the running state of an instruction is followed to a detail by the single step approach. Thus, the trace approach is indispensable although a system or a program is debugged efficiently.

35 [0008] Next, a conventional microprocessor with a built-in cache and a conventional trace system are explained with reference to drawing 22 and drawing 23 . As shown in drawing 22 , including an instruction decoder (IDU) 11, this instruction decoder 11 decodes the instruction code from an instruction cache 12, and the conventional microprocessor 1 sends out various control signals to EU (EXU) 13. Consequently, EU 13 performs various processings according  
40 to these control signals. EU 13 sends out the address (I. ADD) of the instruction which should

be executed next to coincidence at an instruction cache 12. Or EU 13 can send out the instruction address to the bus control unit (BCU) 14 through internal address bus IAB, and can also send out this address to external memory 2 (it does not illustrate to drawing 22 but illustrates to drawing 23 ) through the external address bus AB. In this case, the bus control unit 14 will send out the status signal which shows an instruction fetch cycle to the status signal line ST, consequently external memory 2 will be accessed in the above-mentioned address. Next, it will be received by the bus control unit 14 and the instruction code from external memory 2 will be written in an instruction cache 12 through internal data bus IDB.

[0009] The conventional trace system for the microprocessor 1 of drawing 22 accompanied by external memory 2 contains the marker 3 which consists of a trace analyzer 31 and trace memory 32, as shown in drawing 23 . The trace analyzer 31 supervises status-line ST, and detects an instruction fetch cycle. The trace memory 32 receives the trace address (T. ADD) from the trace analyzer 31, and forms trace of an instruction.

[0010] First, actuation of the marker 3 when the instruction cache 12 is not operating is explained. That is, if an instruction fetch cycle occurs in a microprocessor 1, a microprocessor 1 sends out the address of the instruction which should be carried out a fetch to the external address bus AB while sending out the status signal which starts an instruction fetch cycle and shows the cycle to the status signal line ST. In this condition, when the trace analyzer 31 detects this instruction fetch cycle, the address on the external address bus AB which the trace analyzer 31 sent out strobe signal SR to the trace memory 32, consequently was generated in the instruction fetch cycle will be written in the trace memory 32. Thus, all the addressees of the instruction executed by the microprocessor 1 are recorded on the trace memory 32, and, thereby, can form trace of an instruction.

[0011] Next, actuation of the marker 3 when the instruction cache 12 is operating is explained. Also in this case, EU 13 sends out the address of an instruction to an instruction cache 12. When this instruction is not registered into an instruction cache 12, (it considering as a mistake hit henceforth) and an instruction cache 12 require the fetch of the instruction which carried out the mistake hit of the bus control unit 14. Next, the bus control unit 14 sends out the status signal which starts a replacement cycle on the status signal line ST, and generates in coincidence the address of this instruction that carried out the mistake hit in the external address bus AB. Consequently, the code of the instruction which carried out the mistake hit will be read from external memory 2, and will be registered into an instruction cache 12. Moreover, the code of this instruction that carried out the mistake hit is supplied to an instruction decoder 11 from an instruction cache 12. Also in this case, since the trace analyzer 31 detects an instruction fetch cycle on the status signal line ST, it forms trace of an instruction in the trace memory 32, and can perform the trace approach. Thus, in a microprocessor with built-in cache memory, when the instruction which should be executed is not registered into cache memory (that is, mistake hit), although a fetch is carried out from external memory, consequently the trace approach is performed, it is satisfactory [ an instruction of this mistake hit ] at all.

[0012]



[Problem(s) to be Solved by the Invention] However, when the above-mentioned instruction is registered into the instruction cache 12, (it considering as a hit henceforth) and an instruction cache 12 send out the code of the hit instruction to an instruction decoder 12 immediately. Therefore, since a microprocessor 1 does not generate an external signal at all in this case, a marker 3 can perform the trace approach.

[0013] In addition, in order to perform the trace approach completely, at the time of trace, the approach of enabling observation of an instruction from the exterior of a microprocessor altogether by which the fetch was carried out by making cache memory into non-activity can be considered. However, since it differs from actuation of the microprocessor at the time of actuation with actual actuation of the microprocessor at the time of trace in this case, exact debugging has the technical problem are impossible.

[0014] Therefore, the purpose of this invention is to offer the microprocessor with built-in cache memory which made it possible to perform the trace approach efficiently. Other purposes are to offer the trace system for the above-mentioned microprocessors with built-in cache memory.

[0015]

[Means for Solving the Problem] A branch-instruction detection means detect the branch instruction which should be executed to a microprocessor with a built-in cache according to [ in order to solve an above-mentioned technical problem ] this invention, a notice means of branch-instruction detection notify outside whether branch instruction was detected, the branching formation / abortive distinction means that distinguish whether branching was materialized by activation of branch instruction, and the branching formation / abortive notice means which notify outside whether branching was materialized or not prepare.

[0016] Moreover, the trace system is equipped with the means which blocks a branching place, when the means which traces the following block when the branch instruction detecting signal from a microprocessor, and branching formation / abortive signal are received, and branch instruction is executed and branching carries out failure, and branch instruction are executed and branching is materialized.

[0017]

[Function] In an above-mentioned microprocessor, if branch instruction will be executed even if a cache hits, formation/failure of branch instruction detection and its branch instruction will be notified to the exterior. Consequently, a trace system can trace the instruction execution of the microprocessor at the time of the hit of a cache.

[0018]

[Example] Drawing 1 is the block circuit diagram showing the 1st example of the microprocessor with a built-in cache of this invention. In drawing 1, 15 is an OR circuit and detects the static branch instruction SI and the dynamic branch instruction DI. In addition, although control will shift to that branching place if static branch instruction is executed and branching is materialized, this branching place is beforehand determined in this case. On the other hand, too, if dynamic branch instruction is executed and branching is materialized, although control shifts to that branching place, the branching place in this case is not calculated that is, determined statically beforehand. An example of a dynamic decision

instruction is a register indirect decision instruction. Thus, detecting signal S1 which detects the branch instruction by which OR circuit 15 is performed. It generates and a direct output is carried out to the exterior of a microprocessor 1.

5 [0019] Moreover, 16 is an interruption detection unit which detects internal interruption from dynamic branch instruction and EU 13, and external interruption. EU 13 contains the branch condition distinction machine 134 which distinguishes whether branching was materialized or not as a result of activation of the computing element 131 which generates internal interruption, the branch-address generator 132 which generates a branching place (address),  
10 the interruption address generation machine 133 which generates the interruption address, and branch instruction. That is, if branching is materialized, a selector 135 chooses a branch address and sends it out to an instruction cache 12 as the instruction address. Moreover, if an interrupt occurs, a selector 135 chooses the interruption address and sends it out to an instruction cache 12 as the instruction address. On the contrary, when branching does not produce failure and interruption, either, a selector 135 chooses the output of +1 incrementer  
15 136, and sends it out to an instruction cache 12. +1 incrementer 136 increases the output of a selector 135 +one time, and, thereby, makes sequential access possible.

[0020] Moreover, the branch condition distinction machine 134 is the branching formation / abortive signal S2 which shows whether branching was materialized or not as a result of activation of branch instruction. It generates and sends out to the exterior of a microprocessor  
20 1 directly. Furthermore, EU 13 contains the data register 137 connected to the computing element 131, the branch-address generator 132, the interruption address generation machine 133, and the branch condition distinction machine 134, and the data-access control unit 138 connected between this data register 137 and the bus control unit 14.

[0021] Furthermore, when the interruption detection unit 16 detects dynamic branch  
25 instruction, trace trap processing is started. For example, a program counter (PC) and a program status word (PSW) (not shown) are evacuated first. Next, the interruption detection unit 16 interrupts and sends out interruption to the address generation machine 133, and a selector 135 interrupts, the address generation machine 133 is chosen, and this fetches a trace trap program. In this case, trace to this fetch access is not performed. next, the instruction  
30 cache 12 -- a flash plate -- that is, it is cancelled. Finally, dynamic branch instruction is executed again. In this case, when the branch condition distinction machine 134 distinguishes branching as having been materialized, the branching place (address) of the branch-address generator 132 is chosen by the selector 135, consequently control is jumped at this branching place. In this case, since the instruction cache 12 is cancelled, therefore an instruction cache 12  
35 surely carries out a mistake hit, trace of this branching place address will be performed.

[0022] Drawing 2 is the detailed block circuit diagram of the instruction cache 12 of drawing 1. In drawing 2, an instruction cache 12 contains the tag section 121 and data division 122 which are accessed by the lower bit of the instruction address (I. ADD) supplied from EU 13 at coincidence. In this case, although the tag section 121 memorizes the high-order-bit HB of the  
40 instruction address and another side and data division 122 memorize instruction code, such instruction codes are specified by the instruction address of the high-order-bit HB combination

memorized from EU 13 to the lower bit LB and the tag section 121 of the instruction address. In this case, the high-order-bit HB of the instruction address from EU 13 is not necessarily in agreement with the high-order-bit HB of the tag section 121. Therefore, when the high-order-bit HB of the instruction address from EU 13 is in agreement with the high-order-bit HB of the tag section 121 accessed by the lower bit LB of the instruction address from EU 13, it can be said the instruction cache 12 "hit." On the contrary, when the high-order-bit HB of the instruction address from EU 13 is not in agreement with the high-order-bit HB of the tag section 121 accessed by the lower bit LB of the instruction address from EU 13, it can be said that the instruction cache 12 carried out the "mistake hit."

[0023] In order to distinguish [ to which the instruction cache 12 hit / or or ] whether the mistake hit was carried out, a comparator 123 is formed and it is comparing the high-order-bit HB from EU 13 with the high-order-bit HB from the tag section 121. The hits / misses hit signal of a comparator 123 is supplied to the cache control unit 124. If an instruction cache 12 hits, the cache control unit 124 will send out a control signal to a selector 125, and, thereby, a selector 125 will choose the output of data division 122. Consequently, instruction code will be supplied to an instruction decoder 11 from data division 122. On the contrary, if an instruction cache 12 carries out a mistake hit, the cache control unit 124 controls a latch / gate circuit 126, and sends out the instruction address from EU 13 to the bus control unit 14 while it sends out an instruction access request to the bus control unit 14. Therefore, the bus control unit 14 starts a replacement bus cycle, and fetches instruction code from external memory 2 (refer to drawing 5 ), consequently this instruction code by which the fetch was carried out is written in data division 122. The cache control unit 124 will control a selector 125 to coincidence, and will choose internal data bus IDB as it, and the above-mentioned instruction code by which the fetch was carried out will be supplied to an instruction decoder 11.

[0024] Drawing 3 is the detailed block circuit diagram of the bus control unit 14 of drawing 1 . In drawing 3 , the bus control unit 14 contains the data buffer 141 prepared between internal data bus IDB and an external data bus DB, and the address buffer 142 formed between internal address bus IAB and the external address bus AB. Moreover, an address buffer 142 receives the instruction address from an instruction cache 12. These two buffers 141 and 142 are controlled by the bus access control unit 143, and this bus access control unit 143 receives the control signal from the data access control unit 138 of EU 13, and the control signal from an instruction cache 12. Moreover, the bus access control unit 143 generates a status (ST) signal, a light / lead (W/R) signal, etc. As an example of the instruction code used in drawing 1 is shown in drawing 4 , it consists of the functional field and the register appointed field, and the functional field is further formed of the static branching order bit, the dynamic branching order bit, etc.

[0025] Next, the trace system of the microprocessor of drawing 1 is explained with reference to drawing 5 and drawing 6 . As shown in drawing 5, the trace analyzer S1 is a microprocessor 1 to the detecting signal S1. And branching formation / abortive signal S2 It differs from drawing 23 at the point to receive. Consequently, the trace analyzer S1 is constituted as shown in drawing 6 .

[0026] Setting to drawing 6 , the trace control unit 311 is the detecting signal S1 from a microprocessor 1. And strobe signal SR which receives the control signal from the status decoder 313, and shows starting of trace is generated. For example, if the status signal of the status signal line ST shows the bus cycle of instruction access, the status decoder 313 will generate such a control signal.

[0027] Moreover, it sets to drawing 6 and the flow analysis unit 312 is branching formation / abortive signal S2 from a microprocessor 1. And the control signal from the status decoder 313 is received. The flow analysis unit 312 has memorized the flow data, for example, the address of the beginning of a block, beforehand analyzed before program execution. Moreover, the flow analysis unit 312 is branching formation / abortive signal S2. And the address of a new block is generated using the address of a latch circuit 314, and it sends out to the trace memory 32 by making this into the trace address. The contents of the latch circuit 314 are updated by coincidence by the flow analysis unit 312. In addition, "a block" is the bundle of the instruction executed sequentially here. In other words, neither branching nor unification is into one block.

[0028] Actuation of the combination of the trace control unit 311 and the flow analysis unit 312 is clearly understood by drawing 7 , drawing 8 , and drawing 9 . It shall set to drawing 7 and only the static branch instruction for which a branching place (address) can calculate that is, opt beforehand as branch instruction shall be included. When drawing 7 is referred to, it sets to step 901 and is a detecting signal S1. It detects whether it followed and branch instruction was executed. Only when branch instruction is executed, control shifts to step 902. In other cases, it returns to step 901.

[0029] At step 902, it distinguishes whether according to the output of the status decoder 313, bus access cycles, such as a bus access cycle, for example, an instruction access cycle, and a cache replacement cycle, exist. Consequently, if a bus access cycle exists, control will progress to step 904 and will perform the usual trace. That is, the instruction address is sent out to the trace memory 32 as read in and the trace address from the external address bus AB. In this case, the trace control unit 311 operates a latch circuit 314 at the same time it generates strobe signal SR. In other cases, it progresses to step 903.

[0030] At step 903, they are branching formation / abortive signal S2. It distinguishes whether it followed and branching was materialized. If branching is abortive, it will progress to step 905, and if branching is formation, it will progress to step 906.

[0031] Trace of the address of the beginning of the following block is performed at step 905. That is, since the flow analysis unit 312 is analyzing the instruction flow beforehand like \*\*\*\*, the flow analysis unit 312 recognizes the address of the beginning of each block beforehand. Therefore, for branching, the flow analysis unit 312 is the signal S2 with which an abortive thing is shown in this case. If it receives, the flow analysis unit 312 is sent out to the trace memory 32 through a latch circuit 314 by making the address of the beginning of the following block into the trace address.

[0032] At step 906, trace of the address of the beginning of a branching place block is performed. Namely, if the flow analysis unit 312 receives the signal which shows that branching is formation, the flow analysis unit 312 is sent out to the trace memory 32 by making the address

of the beginning of the branching place block containing a branching place (address) into the trace address.

5 [0033] The control by steps 904 and 905 or 906 returns to step 901. Next, it is assumed that the dynamic branch instruction which cannot calculate that is, determine a branching place (address) beforehand as branch instruction is also included. In this case, in the microprocessor 1 of drawing 1, if a dynamic decision instruction like register indirect instruction is executed, control is jumped to the trace trap routine (processing) shown in drawing 8, and sends out a branching place (address) to the external address bus AB. In this case, the trace analysis unit 312 has analyzed and memorized the branching place. Therefore, the combination of the trace control unit 311 and the flow analysis unit 312 operates like drawing 9.

10 [0034] In drawing 9, steps 1101 and 1102 are added to drawing 7. That is, when it is distinguished that branching was materialized at step 903, it progresses to step 1101, and it distinguishes whether according to the output of the status decoder 313, trace trap processing was performed in the microprocessor 1. If trace trap processing is not performed, it progresses to step 906 and the address of the beginning of the following block is traced. On the contrary, if trace trap processing is performed, it will progress to step 1102.

15 [0035] At step 1102, the flow analysis unit 312 generates the branching place (address) already memorized in trace trap processing, and sends it out to the trace memory 32 through a latch circuit 314 by making this into the trace address. In drawing 9, after distinguishing that branching was materialized at step 903 and tracing the address of the beginning of a branching place block first, if activation of trace trap processing is detected, it is also substitutable with trace of the address output by trace trap processing.

20 [0036] Drawing 10 shows the 2nd example of the microprocessor concerning this invention. In drawing 10, the status encoder 17 is added to the component of drawing 1. This status encoder 17 is a detecting signal S1. And branching formation / abortive signal S2. Since it sends out to the status signal line ST, it uses. Consequently, in constituting a marker to the microprocessor 1 of drawing 10, wiring special between a microprocessor 1 and a marker 3 becomes unnecessary. Therefore, a trace system can be constituted as what is shown in drawing 23, and same thing.

25 [0037] The logic-gate circuit which generates four status bits ST3, ST2, ST1, and ST0 shown in drawing 11 can constitute the status encoder 17. Moreover, the circuit shown in drawing 12 can constitute the trace analyzer 31. That is, as shown in drawing 12, status decoder 313' decodes four status bits ST3, ST2, ST1, and ST0, and sends out a decoding signal to the trace control unit 311, the flow analysis unit 312, and a latch circuit 314. In this case, the combination of the trace control unit 311 of drawing 12 and the flow analysis unit 312 operates, as it operates like the case of drawing 6, therefore the flow chart of drawing 7 and drawing 9 shows.

30 [0038] Drawing 13 shows the 3rd example of the microprocessor concerning this invention. In drawing 13, the cache control register 18 is added to the component of drawing 1. This cache control register 18 is connected to internal data bus IDB, and this distinguishes whether the instruction cache 12 is being activated. That is, the cache control register 18 is cache-on / off signal S3. It generates, and while sending this out to the exterior, it sends out to an instruction

35  
40

cache 12.

[0039] Cache-on signal S3 with which the cache control register 18 activates an instruction cache 12 When it generates, an instruction cache 12 is activated. In this case, if an instruction cache 12 hits, an instruction cache 12 sends out instruction code to an instruction decoder 11.

5 On the contrary, if an instruction cache 12 carries out a mistake hit, an instruction cache 12 is sent out to the bus control unit 14 with the instruction address which carried out the mistake hit of the fetch demand.

[0040] On the other hand, cache off signal S3 with which the cache control register 18 makes an instruction cache 12 non-activity If sent out, an instruction cache 12 will serve as  
10 non-activity. In this case, even if an instruction cache 12 hits, the instruction cache 12 sends out a fetch demand to the bus control unit 14 with the hit instruction address.

[0041] Drawing 14 is the detailed block circuit diagram of the instruction cache 12 of drawing 13 . It sets to drawing 14 and is cache-on / OFF signal S3 of the cache control register 18. The cache control unit 124 is supplied. Cache-on signal S3 with which the cache control register 18  
15 activates an instruction cache 12 When it generates, the cache control unit 124 operates like the case of drawing 2 . On the contrary, cache off signal S3 with which the cache control register 18 makes an instruction cache 12 non-activity When it generates, regardless of the hits / misses hit output of a comparator 123, the cache control unit 124 generates a fetch demand with the instruction address from a latch circuit 126. Moreover, the cache control unit 124  
20 controls a selector 125, and makes internal data bus IDB choose in this case.

[0042] Drawing 15 shows the trace system of the microprocessor 1 of drawing 13 . It sets to drawing 15 and is cache-on / OFF signal S3 of a microprocessor 1. The point currently supplied to the trace analyzer 31 differs from the case of drawing 1 . Consequently, in the trace analyzer 31, as shown in drawing 16 , AND circuits 315 and 316 are added to the component of drawing  
25 6 . Namely, cache-on / OFF signal S3 When "1" (cache ON state) is shown, AND circuits 315 and 316 are enabling, and it is two signals S1 and S2. Making [ consequently ] it pass, the trace analyzer 31 operates like the case of drawing 6 . On the contrary, cache-on / off signal S3 When "0" (cache OFF state) is shown, AND circuits 315 and 316 are disabled, therefore the combination of the trace control unit 311 and the flow analysis unit 312 performs the usual  
30 trace. That is, it is because the microprocessor 1 of drawing 13 operates as a microprocessor without a cache in this case, therefore all instruction accesses are sent out to the exterior of a microprocessor 1.

[0043] Branch instruction operates, as shown in a thing only including static branch instruction then the trace control unit 311 of drawing 16 , the flow analysis unit 312, and  
35 combination drawing 17 of AND circuits 315 and 316. In drawing 17 , step 1901 is added to drawing 7 . That is, at step 1901, they are cache-on / off signal S3. It follows and an instruction cache 12 distinguishes whether it is activation (ON state). Consequently, if the instruction cache 12 is deactivated (off), it will progress to step 904 and the usual trace will be performed. On the contrary, if the instruction cache 12 is activated (ON), it will progress to step 901 and  
40 actuation shown in drawing 7 will be performed after that.

[0044] Moreover, branch instruction operates, as shown in a thing including static branch

instruction and dynamic branch instruction then the trace control unit 311 of drawing 16 , the flow analysis unit 312, and combination drawing 18 of ANI) circuits 315 and 316. In drawing 18 , step 2001 is added to drawing 9 . That is, at step 2001, they are cache-on / off signal S3. It follows and an instruction cache 12 distinguishes whether it is activation (ON state).  
5 Consequently, if the instruction cache 12 is deactivated (off), it will progress to step 904 and the usual trace will be performed. On the contrary, if the instruction cache 12 is activated (ON), it will progress to step 901 and actuation shown in drawing 9 will be performed after that.

[0045] Drawing 19 shows the 4th example of the microprocessor concerning this invention. In drawing 19 , status encoder 17' is added to the component of drawing 13 . This status encoder  
10 17' is a detecting signal S1, the branching formation / abortive signal S2, and the cache-on / off signal S3. Since it sends out to the status signal line ST, it uses. Consequently, in constituting a marker to the microprocessor 1 of drawing 13 , wiring special between a microprocessor 1 and a marker 3 becomes unnecessary. Therefore, a trace system can be constituted as what is shown in drawing 23 , and same thing.

[0046] The logic-gate circuit which generates five status bits ST4, ST3, ST2, ST1, and ST0 shown in drawing 20 can constitute status encoder 17'. Moreover, the circuit shown in drawing 21 can constitute the trace analyzer 31. That is, as shown in drawing 21 , status decoder 313' decodes five status bits ST4, ST3, ST2, ST1, and ST0, and sends out a decoding signal to the trace control unit 311, the flow analysis unit 312, and a latch circuit 314. In this case, the  
20 combination of the trace control unit 311 of drawing 21 and the flow analysis unit 312 operates, as it operates like the case of drawing 14 , therefore the flow chart of drawing 17 and drawing 18 shows.

[0047] In addition, in an above-mentioned example, although the register indirect decision instruction was made into the example as a dynamic decision instruction, this invention is  
25 applicable also to the exception-handling program by activation of exception handling. Moreover, in drawing 7 , drawing 9 , drawing 17 , and drawing 18 , although the address of the beginning of a block is traced, the number representing a block like the block number which is useful to the analysis of an instruction flow may be traced.

[0048]

[Effect of the Invention] Since the information on whether the information and branching about activation of branch instruction were materialized is outputted to the exterior of a microprocessor with a built-in cache according to this invention as explained above, even if an instruction cache hits, trace of an instruction is performed using such information and efficient debugging is enabled.

---

#### DESCRIPTION OF DRAWINGS

---

[Brief Description of the Drawings]

[Drawing 1] It is the block circuit diagram showing the 1st example of the microprocessor

concerning this invention.

[Drawing 2] It is the detailed block circuit diagram of the instruction cache of drawing 1 .

[Drawing 3] It is the detailed block circuit diagram of the bus control unit of drawing 1 .

[Drawing 4] An example of the instruction code used for the microprocessor of drawing 1 is shown.

[Drawing 5] It is the block circuit diagram showing the trace system for the microprocessor of drawing 1 .

[Drawing 6] It is the detailed block circuit diagram of the trace analyzer of drawing 5 .

[Drawing 7] It is the flow chart which shows actuation of the marker of drawing 5 .

[Drawing 8] It is drawing showing trace trap processing in which it operates in the microprocessor of drawing 1 .

[Drawing 9] It is the flow chart which shows actuation of the marker of drawing 5 .

[Drawing 10] It is the block circuit diagram showing the 2nd example of the microprocessor concerning this invention.

[Drawing 11] It is drawing showing the output of the status encoder of drawing 10 .

[Drawing 12] It is the detailed block circuit diagram of the trace analyzer for microprocessors of drawing 10 .

[Drawing 13] It is the block circuit diagram showing the 3rd example of the microprocessor concerning this invention.

[Drawing 14] It is the detailed block circuit diagram of the instruction cache of drawing 13 .

[Drawing 15] It is the block circuit diagram showing the trace system for the microprocessor of drawing 13 .

[Drawing 16] It is the detailed block circuit diagram of the trace analyzer of drawing 15 .

[Drawing 17] It is the flow chart which shows actuation of the marker of drawing 15 .

[Drawing 18] It is the flow chart which shows actuation of the marker of drawing 15 .

[Drawing 19] It is the block circuit diagram showing the 4th example of the microprocessor concerning this invention.

[Drawing 20] It is drawing showing the output of the status encoder of drawing 19 .

[Drawing 21] It is the detailed block circuit diagram of the trace analyzer for the microprocessor of drawing 19 .

[Drawing 22] It is the block circuit diagram showing the conventional microprocessor.

[Drawing 23] It is the block circuit diagram showing the trace system for the microprocessor of drawing 22 .

[Description of Notations]

1 -- Microprocessor

2 -- External memory

3 -- Marker

11 --, an instruction decoder

12 -- Instruction cache

13 -- EU

14 -- Bus control unit



15 -- OR circuit  
 16 -- Interruption detection unit  
 17 17' -- Status encoder  
 18 -- Cache control register  
 5 31 -- Trace analyzer  
 32 -- Trace memory  
 121 -- Tag section  
 122 -- Data division  
 123 -- Comparator  
 10 124 -- Cache control unit  
 125 -- Selector  
 126 -- A latch/gate  
 131 -- Computing element  
 132 -- Branch-address generator  
 15 133 -- Interruption address generation machine  
 134 -- Branch condition distinction machine  
 135 -- Selector  
 136 -- +1 incrementer  
 137 -- Data register  
 20 138 -- Data access control unit  
 311 -- Trace control unit  
 312 -- Flow analysis unit  
 313 313' -- Status decoder  
 314 -- Latch circuit  
 25 315 316 -- AND circuit

---

[Translation done.]

30